

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Tomomichi OBARA, et al.

Application No.:

Group Art Unit:

Filed: April 21, 2004

Examiner:

For: AUTOMATIC TRANSACTION APPARATUS, CONTROL METHOD OF AUTOMATIC TRANSACTION APPARATUS, AND AUTOMATIC TRANSACTION SYSTEM

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55**

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s) herewith a certified copy of the following foreign application:

Japanese Patent Application No(s). 2003-390477


Filed: November 20, 2003

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing date(s) as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: April 21, 2004

By: 
H. J. Staas
Registration No. 22,010

1201 New York Ave, N.W., Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: 2003年11月20日

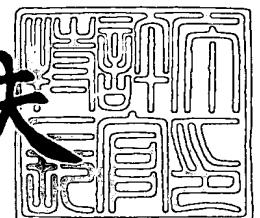
出願番号
Application Number: 特願2003-390477
[ST. 10/C]: [JP 2003-390477]

出願人
Applicant(s): 富士通株式会社
富士通フロンテック株式会社

2004年 1月19日

特許庁長官
Commissioner,
Japan Patent Office

今井康夫



出証番号 出証特2004-3000899

【書類名】 特許願
【整理番号】 0352046
【提出日】 平成15年11月20日
【あて先】 特許庁長官 殿
【国際特許分類】 G07D 9/00
【発明者】
 【住所又は居所】 群馬県前橋市問屋町 1 丁目 8 番地 3 株式会社富士通ターミナル
 システムズ内
 【氏名】 小原 知道
【発明者】
 【住所又は居所】 東京都稲城市矢野口 1 7 7 6 番地 富士通フロンテック株式会社
 内
 【氏名】 富 哲
【特許出願人】
 【識別番号】 000005223
 【氏名又は名称】 富士通株式会社
【特許出願人】
 【識別番号】 000237639
 【氏名又は名称】 富士通フロンテック株式会社
【代理人】
 【識別番号】 100094514
 【弁理士】
 【氏名又は名称】 林 恒徳
【選任した代理人】
 【識別番号】 100094525
 【弁理士】
 【氏名又は名称】 土井 健二
【手数料の表示】
 【予納台帳番号】 030708
 【納付金額】 21,000円
【提出物件の目録】
 【物件名】 特許請求の範囲 1
 【物件名】 明細書 1
 【物件名】 図面 1
 【物件名】 要約書 1
 【包括委任状番号】 9704944
 【包括委任状番号】 0210821

【書類名】 特許請求の範囲**【請求項 1】**

利用者の操作に応じて、Webサーバと通信し、案内表示と取引動作を行う自動取引装置において、

前記案内表示を行う表示ユニットと、

前記取引動作を行うための複数のIOユニットと、

前記Webサーバからの画面コンテンツに応じて、前記表示装置の画面の案内表示を制御し、且つ前記画面コンテンツに埋め込まれたオブジェクトに応じて、前記複数のIOユニットを制御する制御ユニットとを有し、

前記制御ユニットは、前記オブジェクトのスクリプトにより、前記IOユニットをシーケンス制御するとともに、前記表示ユニットの表示文言を前記シーケンス制御に応じて、更新する

ことを特徴とする自動取引装置。

【請求項 2】

前記制御ユニットは、前記Webサーバからの画面コンテンツのアプレットタグを解釈し、前記案内表示を行うとともに、前記画面コンテンツに埋め込まれた前記オブジェクトのアプレットタグを解釈し、前記IOユニットを制御する処理単位のメソッドを呼び出すブラウザを有し、

前記ブラウザから前記IOユニットをシーケンス制御し、且つ前記シーケンス制御に応じて、前記表示ユニットの表示文言を更新する

ことを特徴とする請求項 1 の自動取引装置。

【請求項 3】

利用者の操作に応じて、Webサーバと通信し、案内表示と取引動作を行う自動取引装置の制御方法において、

前記Webサーバからの画面コンテンツに応じて、表示装置の画面の案内表示を行うステップと、

前記画面コンテンツに埋め込まれたオブジェクトに応じて、IOユニットをシーケンス制御するステップと、

前記オブジェクトにより、前記IOユニットのシーケンス制御に応じて、前記表示ユニットの表示文言を更新するステップとを有する

ことを特徴とする自動取引装置の制御方法。

【請求項 4】

前記案内表示ステップは、ブラウザにより、前記Webサーバからの画面コンテンツのアプレットタグを解釈し、前記案内表示を行うステップからなり、

前記制御ステップは、前記ブラウザにより、前記画面コンテンツに埋め込まれた前記オブジェクトのアプレットタグを解釈し、前記IOユニットを制御する処理単位のメソッドを呼び出すステップからなり、

前記更新ステップは、前記ブラウザから前記IOユニットをシーケンス制御したことに応じて、前記表示ユニットの表示文言を更新するステップからなる

ことを特徴とする請求項 3 の自動取引装置の制御方法。

【請求項 5】

Webサーバと、

利用者の操作に応じて、前記Webサーバと通信し、案内表示と取引動作を行う自動取引装置とを有し、

前記自動取引装置は、

前記案内表示を行う表示ユニットと、

前記取引動作を行うための複数のIOユニットと、

前記Webサーバからの画面コンテンツに応じて、前記表示装置の画面の案内表示を制御し、且つ前記画面コンテンツに埋め込まれたオブジェクトに応じて、前記複数のIOユニットを制御する制御ユニットとを有し、

前記制御ユニットは、前記オブジェクトのスクリプトにより、前記 I O ユニットのシーケンス制御するとともに、前記表示ユニットの表示文言を前記シーケンス制御に応じて、更新すること
ことを特徴とする自動取引システム。

【書類名】 明細書**【発明の名称】 自動取引装置、自動取引装置の制御方法及び自動取引システム****【技術分野】****【0001】**

本発明は、リクエストによりWebサーバから送信された画面コンテンツに応じて、画面制御と自動取引動作とを実行する自動取引装置及び自動取引システムに関し、特に、画面情報と画面に関連する装置の制御情報を埋め込んだ画面コンテンツにより動作する自動取引装置及び自動取引システムに関する。

【背景技術】**【0002】**

自動取引装置は、各種の取引に利用されており、例えば、金融分野では、自動出金機、自動入出金機が、他の分野では、自動発券機や自動発行機等が利用されている。このような自動取引装置において、近年のインターネット等のネットワークの発達に伴い、Web技術を利用して、入出金、発券、各種情報の出力を行うものが提供されている。

【0003】

図27は、従来の自動取引システムの構成図であり、金融業務のATM (Automatic Teller Machine) システムを示す。図27に示すように、WWW (World Wide Web) サーバ300と、現金自動取引装置400とがネットワークで接続される。

【0004】

サーバ300は、クライアントである自動取引装置(ATM)400のリクエストに従い、Webページ(画面コンテンツ)500をATM400に送信する。このWebページは、表示装置に表示される画面を作成するためのプログラムを、ページ記述言語(HTML, Java (登録商標) Script)を用いて作成し、この1ページ(1画面)の表示内容に関連して、駆動制御する他の装置(カード処理装置、現金処理装置、通帳処理装置、明細票処理装置等)の制御プログラムをオブジェクトとして、埋め込む。

【0005】

例えば、図27に示すように、Webページ500は、ページ記述言語HTML (Hyper Text Markup Language)により、画面作成プログラムと、オブジェクト(アプレット)を指定するアプレットタグと、その実行方法(method)をScriptで指定する画面コンテンツ502と、オブジェクト(アプレット)のmethodを実行するプログラムをセットしたアプレット510からなる。

【0006】

このWebページ500は、ATM400のリクエストに従い、WWWサーバ300からATM400のブラウザ410にダウンロードされる。一方、ATM400は、カーネル(OS)の制御の基に、ATMミドルウェアが動作し、I/O動作(取引動作)する。ATM400は、I/Oメカユニットとして、カードリーダー/ライターユニット440、レシート/ジャーナルプリンタ441、紙幣/硬貨処理ユニット442、通帳処理ユニット443、顧客操作パネル等を有する。

【0007】

ブラウザ410は、Webページの画面作成プログラムに従い、画面を顧客操作パネルに表示するとともに、画面コンテンツ502のアプレットタグと、method名を解析し、アプレット510の該当プログラムを実行し、I/Oユニット440～443にコマンドを発行する。

【0008】

このようなWebブラウザによるATM制御において、図27及び図28に示すように、methodを、各装置(現金処理装置等)単位とし、画面コンテンツ502の埋め込みオブジェクト(アプレット)として、装置インタフェース振分け部(Machine ID)を指定し、画面コンテンツ502のScript (Java script)で、各装置の動作(この例では、Initialize)方法を指定することが提案されている(例えば、特許文献1参照)。

【0009】

この方法では、画面コンテンツ 502 のアプレットタグで、装置インタフェース振分け部 420 が指定され、そして、装置振分け部 420 は、Script を順次読み込み、解読し、動作指令を対応するインタフェース部 430 ～ 433 に振分け、対応する I/O ユニット 440 ～ 443 を動作する。

【0010】

この例では、装置初期化の動作指令であるから、図 29 に示すように、画面コンテンツ 502 の Script の Initialize("Cash") を読み込み、現金処理装置 442 のコントローラに初期化コマンドを送り、初期化完了応答を得て、次に、Script の Initialize("Card") を読み込み、カード処理装置 440 のコントローラに初期化コマンドを送り、初期化完了応答を得る。以下、順次、Script の Initialize () を読み込み、対応する処理装置のコントローラに初期化コマンドを送り、初期化完了応答を得て、終了する。

【0011】

この提案では、同一機能の ATM を Web 制御する場合に、画面コンテンツの 1 つのアプレットタグで、複数のユニットを動作することができ、画面コンテンツの HTML 内の記述を少なくでき、埋め込みオブジェクトを軽くできる。

【特許文献 1】特開 2000-298752 号公報 (図 6、図 7)

【発明の開示】

【発明が解決しようとする課題】

【0012】

しかしながら、Web 制御では、動的コンテンツを実現するため、画面を変更する毎に、ATM から Web サーバにリクエストを要求する必要がある。自動取引処理では、利用者が複数の操作を行う必要があるため、その操作毎の案内表示が必要である。

【0013】

このため、従来技術では、1 取引に必要な ATM からのリクエスト回数が多く、これに伴い、Web サーバとの通信回数も増加し、通信時間や通信費用が増加するおそれがある。又、通常、Web サーバは、多数の ATM を制御するため、Web サーバの負荷が増大し、且つ ATM (クライアント) でのダウンロード待ち時間が長くなり、取引処理に時間がかかるおそれがある。

【0014】

更に、ATM は、同一の機能のものならず、異なる機能の ATM も多く存在する。例えば、現金処理機能として、紙幣及び硬貨を処理するものと、紙幣のみを処理するものがあり、又、通帳処理を行うものと、行わないものがあり、更に、入出金処理するものと、出金専用のものがある。

【0015】

このような異なる機能の ATM を Web 制御する場合には、この従来技術では、呼び出される method が、各ユニット単位であるため、構成機能又はユニットに応じて、同一のアプレットタグでも、画面コンテンツの Script 自体を大幅に変更する必要がある。即ち、画面コンテンツの記述を、制御する ATM の構成機能又はユニットの相違に応じて、設計する必要がある。

【0016】

このため、WWW サーバ 300 での、Web ページの開発に多大の労力を要し、且つ新しい機能 (例えば、携帯電話との関係機能等) を付加するのに、手間がかかるおそれがある。

【0017】

従って、本発明の目的は、1 取引に必要なリクエスト回数を減少する自動取引装置、自動取引装置の制御方法及び自動取引システムを提供することにある。

【0018】

又、本発明の他の目的は、アプレットで、Web サーバにリクエストを発する必要がない処理を行い、通信時間を減少し、自動取引に要する時間を低減するための自動取引装置、自動取引装置の制御方法及び自動取引システムを提供することにある。

【0019】

又、本発明の他の目的は、Webサーバにリクエストを発する必要がない場合に、アプレットで処理を行い、リクエスト回数を減少するための自動取引装置、自動取引装置の制御方法及び自動取引システムを提供することにある。

【0020】

更に、本発明の他の目的は、アプレットで画面更新処理を行い、リクエスト回数を減少するための自動取引装置、自動取引装置の制御方法及び自動取引システムを提供することにある。

【課題を解決するための手段】**【0021】**

この目的の達成のため、本発明の自動取引装置は、利用者の操作に応じて、Webサーバと通信し、案内表示と取引動作を行う自動取引装置において、前記案内表示を行う表示ユニットと、前記取引動作を行うための複数のIOユニットと、前記Webサーバからの画面コンテンツに応じて、前記表示装置の画面の案内表示を制御し、且つ前記画面コンテンツに埋め込まれたオブジェクトに応じて、前記複数のIOユニットを制御する制御ユニットとを有し、前記制御ユニットは、前記オブジェクトのスクリプトにより、前記IOユニットをシーケンス制御するとともに、前記表示ユニットの表示文言を前記シーケンス制御に応じて、更新する。

【0022】

本発明の自動取引装置の制御方法は、利用者の操作に応じて、Webサーバと通信し、案内表示と取引動作を行う自動取引装置の制御方法において、前記Webサーバからの画面コンテンツに応じて、表示装置の画面の案内表示を行うステップと、前記画面コンテンツに埋め込まれたオブジェクトに応じて、IOユニットをシーケンス制御するステップと、前記オブジェクトにより、前記IOユニットのシーケンス制御に応じて、前記表示ユニットの表示文言を更新するステップとを有する。

【0023】

本発明の自動取引システムは、Webサーバと、利用者の操作に応じて、前記Webサーバと通信し、案内表示と取引動作を行う自動取引装置とを有し、前記自動取引装置は、前記案内表示を行う表示ユニットと、前記取引動作を行うための複数のIOユニットと、前記Webサーバからの画面コンテンツに応じて、前記表示装置の画面の案内表示を制御し、且つ前記画面コンテンツに埋め込まれたオブジェクトに応じて、前記複数のIOユニットを制御する制御ユニットとを有し、前記制御ユニットは、前記オブジェクトのスクリプトにより、前記IOユニットをシーケンス制御するとともに、前記表示ユニットの表示文言を前記シーケンス制御に応じて、更新する。

【0024】

又、本発明では、好ましくは、前記制御ユニットは、前記Webサーバからの画面コンテンツのアプレットタグを解釈し、前記案内表示を行うとともに、前記画面コンテンツに埋め込まれた前記オブジェクトのアプレットタグを解釈し、前記IOユニットを制御する処理単位のメソッドを呼び出すブラウザを有し、前記ブラウザから前記IOユニットをシーケンス制御し、且つ前記シーケンス制御に応じて、前記表示ユニットの表示文言を更新する。

【0025】

又、本発明では、好ましくは、前記制御ユニットは、前記オブジェクトのアプレットと前記画面のアプレットとの通信を行い、前記表示ユニットの表示文言を更新する。

【0026】

又、本発明では、好ましくは、前記制御ユニットは、前記オブジェクトのアプレットにより、前記スクリプトの関数を呼び出し、前記表示ユニットの表示文言を更新する。

【0027】

又、本発明では、好ましくは、前記制御ユニットは、前記画面コンテンツに埋め込まれたオブジェクトのスクリプトに応じて、前記複数のIOユニットを同期制御する処理単位

のメソッドを呼び出し、前記複数の I O ユニットを同期制御する。

【0028】

又、本発明では、好ましくは、前記 I O ユニットは、少なくとも現金を受け入れ、受け入れた現金を鑑別して、入金する現金ユニットを含み、前記制御ユニットは、前記現金ユニットの動作に応じて、前記画面コンテンツのスクリプトを解釈し、前記表示ユニットの表示文言を更新する。

【0029】

又、本発明では、好ましくは、前記制御ユニットは、初回入金と追加入金とを実行した後、前記 Web サーバにリクエストを送信する。

【発明の効果】

【0030】

本発明では、画面の更新を全て Web サーバからダウンロードするのではなく、画面コンテンツのアプレットとスクリプトで、画面の一部を更新するため、Web サーバとの通信を行わずに、画面の更新が可能となる。

【0031】

このため、Web サーバ 100 へのリクエスト回数が減少し、通信回数及び通信時間を減少でき、自動取引装置の取引性能（速度）を向上でき、且つ Web サーバの負荷を軽減できる。

【発明を実施するための最良の形態】

【0032】

以下、本発明の実施の形態を、自動取引システム、Web 制御による I O 制御機構、Agent を用いた自動取引処理、Agent によるリクエスト回数の削減方法、リクエスト回数を削減した自動取引処理、他の実施の形態の順で説明する。

【0033】

〔自動取引システム〕

図 1 は、本発明の一実施の形態の自動取引装置の外観図、図 2 は、図 1 の自動取引装置のブロック図、図 3 は、本発明の一実施の形態の自動取引システムのシステム構成図である。

【0034】

図 1 に示すように、自動取引装置 1 は、磁気カードの挿入及び排出のためのカード出入口 2、磁気通帳の挿入及び排出のための通帳出入口 3、紙幣の投入及び排出のための紙幣出入口 4、硬貨の投入及び排出のための硬貨出入口 5、利用者の操作のための U O P (User Operation Panel) 6、利用者に運用状況を表示するための運用表示器 7、利用者を検出する顧客センサ 8 を備える。

【0035】

図 2 は、図 1 の自動取引装置 1 のブロック図である。C R W (Card Reader Writer) ユニット 10 は、カード出入口（カード挿入口）2 から挿入された磁気カードを、図示しない搬送機構で搬送しながら、磁気ヘッドで読み取り、出入口 2 へ返却する。C R W ユニット 10 には、イメージセンサが設けられ、磁気カード（エンボス部）を光学的に読み取る。

【0036】

R P R (Receipt Printer) ユニット 20 は、印字ヘッドで、レシート用紙に、取引結果の印字を行い、レシートをカード出入口 2 に排出する。又、R P R ユニット 20 は、排出されたレシートが所定時間内に利用者に抜き取られない時に、出入口 2 から戻されたレシートを收容する。

【0037】

J P R (Journal Printer) ユニット 70 は、印字ヘッドで、ジャーナル用紙に、取引状態、結果等を印字する。U O P (User Operation Panel) ユニット 30 は、U O P (タッチパネル付きディスプレイ) 6 とその制御回路で構成される。通帳 (P P R) 処理ユニット 40 は、通帳出入口 3 から挿入された磁気通帳を読み取り、且つ磁気通帳に取引印字

を行い、通帳出入口3から排出する。

【0038】

紙幣／硬貨処理ユニット50は、紙幣出入口4、硬貨出入口5から投入された紙幣、硬貨を鑑別、計数し、スタッカに収容する入金動作を行い、且つ現金スタッカから要求された紙幣、硬貨を取り出し、紙幣出入口4、硬貨出入口5へ放出する出金動作を行う。

【0039】

制御部60は、LAN等のネットワーク90を介し、これら制御ユニット10、20、30、40、50、70に接続され、図3にて後述するソフトウェアの構成により自動取引処理を行う。

【0040】

図3は、本発明の一実施の形態の自動取引システムのブロック図である。自動取引装置1は、インターネット等のネットワーク110を介し、WWW (World Wide Web)サーバ(ホスト)100と、取引処理に必要なコマンド、パラメータ、データのやりとりを行う。

【0041】

自動取引装置1では、前述の制御部60が、ブラウザ120と、ATMミドルウェア130と、カーネル(OS)140と、デバイスドライバ150とを搭載する。

【0042】

デバイスドライバ150は、カード(CRW)ユニット10をドライブするためのカードユニットドライバ151と、レシート／ジャーナルユニット(RPR/JPR)20、70をドライブするためのレシート／ジャーナルユニットドライバ152と、BRU(紙幣)ユニット50をドライブするためのBRUドライバ153と、CRU(硬貨)ユニット50をドライブするためのCRUドライバ154と、UOP30をドライブするためのグラフィックドライバ155及びタッチスクリーンドライバ156と、通帳ユニット40をドライブするためのPPRドライバ157とを有する。

【0043】

ブラウザ120は、例えば、Internet Explorer (Microsoft社商標名)等のWebブラウザで構成され、Webサーバ100にコンテンツの送信を要求し、Webサーバ100が送信する画面コンテンツ(Webページ)を解釈し、表示等を行う。ここでは、HTMLとJava Scriptで構成された取引処理に必要なWebページを要求し、送信されたWebページを解釈し、ATMミドルウェア130、UOP30の画面制御を行う。

【0044】

カーネル140は、例えば、Windows (登録商標)、Linux等の周知のOS (Operating System)で構成され、カーネル140の動作環境下で、ブラウザ120、ATMミドルウェア130、デバイスドライバ150が動作する。

【0045】

ATMミドルウェア130は、パラメータファイル160と、IOコントロール層170と、IOクライアント層190と、IOサーバ層200と、IOサービスプロバイダ層210とを有する。

【0046】

このIOクライアント層190は、装置に実装された個々のI/Oユニットを制御するものであり、IOサーバ層200は、IO動作の起動、終結及び通信プロトコル制御を行うものであり、IOサービスプロバイダ層210は、各IOユニットへの電文の変換を行うものである。これらは、その装置の機能範囲、種類、接続されたIOユニットの仕様に従い、設計された既存のミドルウェア180である。

【0047】

一方、IOコントロール層170は、Webサーバ100とミドルウェア共通アプリケーションインタフェース規約で、コマンド、データの送受信を行う。この共通アプリケーションインタフェース(API)は、装置の機種によって、機能範囲が異なるため、全ての機種を動作できる共通のコマンド、データ体系で構成され、図6で後述する。

【0048】

IOコントロール層170は、IOクライアント層190のアプリケーションインタフェース（API）を統合、集約し、抽象度の高い共通APIを構築するものである。パラメータファイル160は、ベンダ（ATMメーカ）固有のシステム仕様によって、一意に決定される入力パラメータ／固定パラメータを格納するものである。

【0049】

IOコントロール層170は、IOクライアント層190の呼び出し時に、このパラメータファイル160から各IOクライアント層固有のパラメータを呼び出し、共通APIを、既存のクライアントAPIに変換する。

【0050】

これにより、抽象度の高い共通APIを、自動取引装置1のATMミドルウェア190、搭載IOユニットの種類に合致したクライアントAPIに変換でき、既存のATMミドルウェア180とIOユニットを動作することができる。即ち、既存のATMミドルウェアを、共通APIで動作するように、カスタマイズ化できる。

【0051】

図3に示すように、本発明では、Webページの画面コンテンツ122のアプレットとして、ATMの処理単位で定義したエージェント（Agent）123を埋め込み、処理単位で処理に必要なIOユニットの動作を制御する。

【0052】

詳細は、図4以下で後述するが、アプレット名を、同期制御や、同期制御の初期化制御という処理単位で定義し（以下、Agentという）、methodも処理単位で設ける。即ち、処理に対応したIOユニットの動作を制御するmethodを有するアプレット（Agent）を設け、処理単位で、アプレット（Agent）、methodを指定する。

【0053】

これにより、複数ユニットを同期動作する場合でも、処理単位にくることができるため、異なる機能のATMのWeb制御をWebページの変更箇所が少なくでき、且つ複雑な自動取引制御においても、Webページを容易に作成できる。

【0054】

又、1つのmethodをコールするだけで、複数のIOユニットを制御できるため、複数のIOユニットを並行制御でき、それだけ、処理速度を向上でき、Web制御しても、ATMの制御時間を短縮でき、利用者の待ち時間を短縮できる。

【0055】

[Web制御によるIO制御機構]

次に、Web制御によるIO制御機構を説明する。図4は、図3の構成におけるWebブラウザとIOコントローラのブロック図、図5は、図3の構成におけるATMミドルウェアの構成図、図6は、図3及び図4における共通APIのコマンドの説明図である。

【0056】

図4に示すように、Webブラウザ120には、画面コンテンツ（HTML+JavaScript）122と、Agent（アプレット）123とが、WWWサーバ100からダウンロードされる。Agent 123は、各ユニットを制御するための制御Agent群124と、ポスト処理を行うPOST Agent 126と、画面構築のためのView Agent 128と、Java Native Interfaceと共通APIインタフェース132を介しIOコントローラ170と通信するための通信DLL群（Dynamic Link Library）129を有する。

【0057】

尚、インタフェースが同一であれば、通信DLL群129は、不要である。制御Agent群124は、表示画面に必要な制御Agentのみダウンロードされる。図7以下で詳細に説明するように、制御Agentは、ATM取引の処理単位に構成されている。

【0058】

図4の構成の動作を説明すると、画面コンテンツ122のJava Scriptで記述されたメソッド名で、制御Agent124のメソッドを呼び出す。呼び出されたメソッドは、通信DLL群129を介しIOコントローラ170にコマンドを発行する。

【0059】

IOコントローラ170は、ATMミドルウェア180を介し対応するユニットを動作し、動作完了を受ける。IOコントローラ170は、コマンドの実行結果を制御Agent124に応答する。制御Agent124は、Post処理を行うPOSTAgent126又は画面更新を行うViewAgent128にPOST依頼又は描画依頼を依頼し、POST処理又は画面更新処理を行わせる。

【0060】

このAgentを説明する前に、先ず、共通APIを説明する。図6は、共通APIのコマンド種類の一例を示す。CRW（カードリーダー/ライター）コマンドとしては、カード挿入コマンド、カード排出コマンドが用意されている。

【0061】

RPR（レシートプリンタ）コマンドとしては、印字コマンド、放出コマンド等が用意されている。PPR（通帳プリンタ）コマンドとしては、通帳挿入コマンド、印字コマンド、MS（磁気ストライプ）書き込みコマンド、通帳排出コマンド、オートターンページコマンド等が用意されている。

【0062】

BRU（紙幣リサイクルユニット）コマンドとしては、初期化コマンド、受付/計数コマンド、収納コマンド、入金返却コマンド、繰り出しコマンド、放出コマンド、取り込みコマンド、搬送路チェックコマンド、ジャムリセットコマンド等が用意されている。CRU（硬貨リサイクルユニット）コマンドも同様であり、説明を省略する。

【0063】

次に、図3及び図4のATMミドルウェア130の構成を、図5により説明する。IOコントロール層170は、各IOのコントロールを行うIOコントロールライブラリ群（IOコントローラ）171～178を有する。

【0064】

ここでは、IOコントロールライブラリ群は、通帳コントロールライブラリ171、CRWコントロールライブラリ172、テンキーコントロールライブラリ173、レシートコントロールライブラリ174、紙幣コントロールライブラリ175、硬貨コントロールライブラリ176、ジャーナルコントロールライブラリ177、取引制御コントロールライブラリ178を有する。

【0065】

これらコントロールライブラリ171～178は、共通APIで指定されたタスク（カードコントロールEXE等）により呼び出され、そのタスクを前述のパラメータテーブル160を利用して、既存ミドルウェアのクライアントAPIに変換する。

【0066】

又、既存ミドルウェア180のIOクライアント層190は、装置に実装された個々のI/Oユニットを制御するものであり、ここでは、カード（CRW）クライアント191、硬貨クライアント192、紙幣クライアント193、RPRクライアント194、JPRクライアント195、PPRクライアント196等が設けられている。

【0067】

同様に、IOサーバ層200も、個々のIO動作の起動、終結及び通信プロトコル制御を行うため、個々のIOに分割されている。即ち、カード（CRW）サーバ201、硬貨サーバ203、紙幣サーバ202、RPR（レシートプリンタ）サーバ204、JPRサーバ205、PPR（通帳プリンタ）サーバ206が設けられている。

【0068】

又、同様に、IOサービスプロバイダ層210も、各IOユニットへの電文の変換を行うため、個々のIOに分割されている。即ち、カード（CRW）サービスプロバイダ21

1、硬貨サービスプロバイダ213、紙幣サービスプロバイダ212、RPRサービスプロバイダ214、JPRサービスプロバイダ215、PPRサービスプロバイダ216が設けられている。

【0069】

即ち、ATMミドルウェアを構成する、コントロールライブラリ、クライアント、サーバ、サービスプロバイダが、個々のIOユニットに対応して設けられ、要求された共通APIのコマンド、パラメータを、IOコントロール170が、既存ミドルウェアAPIのコマンド、パラメータに変換し、既存ミドルウェアを介しIOユニットを動作する。

【0070】

次に、前述のAgentを、図7、図8で説明する。図7に示すように、同期制御Agentは、複数のIOユニットを同期制御するプログラムである。この例では、Agentとメソッドとを一体にしてあり、同期制御Agent/メソッド（プログラム）として、初期化、メカリセット、紙幣/硬貨挿入、媒体同時排出、印字/繰り出し/MSライト/排出準備、入金返却、収納、強制排出/取込み、ユニット情報取得/取引状態設定/二面表示制御、入出金準備、強制補充、ジャムリセット、カード/通帳挿入等を有する。

【0071】

各同期制御Agent/メソッドは、図7の黒丸で示すIOコントローラ（即ち、IOユニット）にコマンドを発行し、コマンド実行結果の応答を受ける。例えば、同期制御の初期化Agent/メソッドは、紙幣コントローラ175、硬貨コントローラ176、通帳コントローラ171、カードコントローラ172、レシートコントローラ174、ジャーナルコントローラ177、取引制御コントローラ178、テンキーコントローラ173（図5参照）に、初期化コマンドを発行し、各コントローラによる初期化処理を行わせ、各コントローラから初期化処理結果を応答として、受信する。

【0072】

同様に、同期制御のメカリセットAgent/メソッドは、紙幣コントローラ175、硬貨コントローラ176、通帳コントローラ171、カードコントローラ172、レシートコントローラ174、ジャーナルコントローラ177（図5参照）に、メカリセットコマンドを発行し、各コントローラによるメカリセット処理を行わせ、各コントローラからメカリセット処理結果を応答として、受信する。

【0073】

以下、同様に、同期制御の紙幣/硬貨挿入Agent/メソッドは、紙幣コントローラ175、硬貨コントローラ176（図5参照）に、挿入コマンドを発行し、各コントローラによる挿入処理を行わせ、各コントローラから挿入処理結果を応答として、受信する。

【0074】

同様に、POSTエージェントとして、POST処理メソッドと、POSTデータ保持メソッドを設ける。又、View（テキスト表示）エージェントとして、フォント設定メソッド、テキスト表示メソッド、テキスト消去メソッドを設ける。

【0075】

更に、図8に示すように、単一のIOユニットを制御するアプレットも、同一のアーキテクチャで制御するため、Agentとして、定義する。即ち、図8に示すように、紙幣制御Agent、通帳制御Agent、カード制御Agent、レシート制御Agent、取引制御Agent、テンキーAgentを設ける。

【0076】

そして、紙幣制御Agentは、各々紙幣コントローラ175を制御する受付/計数メソッド、収納メソッド、入金返却メソッド、放出メソッド、キャンセルメソッドを有する。又、通帳制御Agentは、各々通帳コントローラ171を制御する行セット/ページマークリードメソッド、MS（磁気ストライプ）リードメソッド、オートターンページメソッド、ページチェックオートターンメソッド、通帳構成情報設定メソッドを有する。

【0077】

更に、カード制御Agentは、各々カード（CRW）コントローラ172を制御する

カード挿入メソッド、キャンセルメソッド、振込みカード印字メソッド、振込みカード発行メソッド、排出準備メソッドを有する。レシート制御Agentは、各々レシートコントローラ174を制御するオーバーレイ登録メソッド、排出準備メソッドを有する。

【0078】

同様に、取引制御Agentは、各々取引制御コントローラ178を制御する取引情報設定メソッド、装置状態監視メソッド、装置状態取得メソッド、運用情報設定メソッド、キャンセルメソッドを有する。テンキーAgentは、各々テンキーコントローラ173を制御するテンキー入力開始メソッド、テンキー入力終了メソッドを有する。

【0079】

このようなAgent及びメソッドを呼び出すWeb操作を、図9乃至図12で説明する。図9は、Webサーバ100が、送信する画面コンテンツの説明図であり、図10は、図9の画面コンテンツによるIO制御の説明図である。

【0080】

図9に示すように、HTML（ページ記述言語）で記述された画面コンテンツ内で、画面内で呼び出すAgent名を指定する。ここでは、一例として、アプレットタグ<APPLET CODE="U__agtSync__initial.class">により、同期制御（Sync）の初期化エージェントを呼び出すことを示す。

【0081】

更に、<SCRIPT Language="JavaScript">の記述により、JavaScriptで、同期制御の初期化エージェントのメソッドの呼び出しを指定する。即ち、ret=document.U__agtSync__initialの記述で、同期制御の初期化エージェント／メソッドの呼び出しを指定する。尚、ここでは、画面表示コンテンツの詳細を省略してある。

【0082】

この画面コンテンツが、ブラウザ120にダウンロードされると、図10に示すように、アプレットタグにより、同期制御エージェントが指定され、且つSCRIPTの呼び出しメソッド名により、初期化メソッドが呼び出される。

【0083】

前述のように、呼び出された初期化メソッドは、紙幣コントローラ175、硬貨コントローラ176、通帳コントローラ171、カードコントローラ172、レシートコントローラ174、ジャーナルコントローラ177、取引制御コントローラ178、テンキーコントローラ173（図5参照）に、初期化コマンドを発行し、各コントローラによる初期化処理を行わせ、各コントローラから初期化処理結果を応答として、受信する。

【0084】

この場合に、図12に示すように、呼び出された初期化メソッドは、紙幣コントローラ175、硬貨コントローラ176、通帳コントローラ171、カードコントローラ172、レシートコントローラ174、ジャーナルコントローラ177、取引制御コントローラ178、テンキーコントローラ173（図5参照）に、初期化コマンドを連続的に発行し、これらコントローラからの応答を、処理完了したものから順次受ける。

【0085】

このため、IOユニットを並列的に制御でき、複数のIOユニットを制御しても、短時間で制御できる。これに比し、図29で示したように、従来のScriptで、IOユニット単位のメソッドを指定する方法では、Script毎に、コマンド発行、応答受信を順次行う順次制御のため、WebによるIO制御に時間がかかる。

【0086】

更に、前述のAgentは、処理単位名が付された処理単位で設けられるため、この処理単位の内容を若干変更するだけで、各種の機能のATMに、同一のアプレットタグ、メソッド名で、対応できる。この例では、前述の図9のメソッド呼び出しSCRIPTのメソッド名の括弧内に設けられたinitial（input Param）の変更により実現している。即ち、図11に示すように、input Paramは、各IOコントローラ（紙幣、硬貨、通帳、カード、レシート、ジャーナル、取引制御、テンキー）に、入力フラグを設定する複数バイ

トのビット情報(図では、8ビットの情報)で構成されている。

【0087】

入力フラグは、“1”がセットされたI/Oコントローラへのメソッドの入力を許可することを示す。呼び出されたメソッドは、この入力フラグを参照して、コマンドを発行するI/Oコントローラを決定する。このため、入力フラグをWebサーバ100で操作することにより、異なる機能のATMでも、同一のアプレットタグ、メソッド名で、同一の処理を実行させることができる。

【0088】

例えば、硬貨を扱わない取引装置では、硬貨の入力フラグを“0”にセットしておけば、硬貨コントローラへのコマンド発行を防止できる。同様に、通帳を扱わない取引装置では、通帳の入力フラグを“0”にセットしておけば、通帳コントローラへのコマンド発行を防止できる。

【0089】

又、SCRIPTのメソッド名の括弧内に設けられたParseInt(postMode)により、POSTエージェントを、このエージェント内で指定している。このため、POST処理を、エージェント主導で円滑に実行できる。又、エージェントが全てのコマンド完了を受信したタイミングで、直ちに、POSTすることができる。

【0090】

更に、同様に、括弧内に設けられたparseInt(maxTimer)により、タイムアウト処理を指定しており、Agentにてタイマー監視機能を持たせ、Webサーバ100との通信無しのタイムアウト時に処理を中断する。

【0091】

[自動取引処理]

次に、図13のカード入金取引を例に、Webサーバ100の画面コンテンツと、顧客操作画面(UOP画面)と、Agentとメソッド、I/Oコントローラとの関係を説明する。

【0092】

Webサーバ100は、JSP(Java Server Pages)により取引種別選択の画面コンテンツ及びアプレットを、ATM1に発行する。ATM1では、ブラウザ120で、取引種別選択画面を、UOP6の顧客操作画面に、表示する。

【0093】

そして、画面コンテンツに埋め込まれたアプレット名とメソッド名により、図7の同期制御Agentの入出金準備メソッドを呼び出し、紙幣コントローラ175と、硬貨コントローラ176に、入出金準備コマンドを発行し、入出金の準備を行わせる。両コントローラ175、176は、入出金の準備が完了すると、入出金準備メソッドに完了を応答する。

【0094】

UOP6の種別選択画面で、入金キーが押下されると、Webサーバ100に報告される。これにより、Webサーバ100は、JSPにより、カード開始処理に移行し、挿入処理の画面コンテンツをATM1に発行する。ATM1では、ブラウザ120で、UOP6の顧客操作画面に、カード挿入画面を表示する。

【0095】

そして、画面コンテンツに埋め込まれたアプレット名カード制御、メソッド名カード挿入により、カード制御Agentのカード挿入メソッドを呼び出し、カードコントローラ172にカード挿入コマンドを発行する。コントローラ172は、カード挿入コマンドにより、カードユニット10を動作する。カードユニット10がカードの挿入を検出し、カードを読取ると、カードコントロールライブラリ172より、カード挿入完了を、カード挿入メソッドに応答する。

【0096】

そして、カード挿入メソッドは、取引状態(ここでは、カード挿入検出及びカードの読

取りデータ)をPOST依頼し、POSTエージェント126が、Webサーバ100にリクエスト送信する。この場合、同期制御Agentのカード/通帳挿入メソッドを呼び出し、その入力フラグ(図11参照)で、カードコントローラに入力フラグを設定しておくことにより、同様に、制御できる。

【0097】

次に、Webサーバ100のJSPは、カード挿入処理の終了によるリクエストにより、現金挿入処理の画面コンテンツをATM1に発行する。ATM1では、ブラウザ120が、画面コンテンツに応じて、UOP6の顧客操作画面に、現金挿入画面を表示する。そして、画面コンテンツに埋め込まれたアプレット名であるカード制御Agent、メソッド名であるカード挿入により、同期制御Agentの紙幣/硬貨挿入メソッドを呼び出し、紙幣コントローラ175、硬貨コントローラ176に受付計数コマンドを発行する。

【0098】

紙幣コントローラ175、硬貨コントローラ176は、受付計数コマンドにより、紙幣ユニット、硬貨ユニット50を動作する。紙幣、硬貨ユニット50は、挿入口を開き、投入紙幣、硬貨を検出し、挿入口を閉じ、投入紙幣、硬貨を計数する。計数を終了すると、紙幣コントローラ175と硬貨コントローラ176より、受付計数完了を、紙幣/硬貨挿入メソッドに応答する。そして、紙幣/硬貨挿入メソッドは、取引状態(ここでは、受付枚数、合計額)をPOST依頼し、POSTエージェント126が、Webサーバ100にリクエスト送信する。

【0099】

次に、Webサーバ100のJSPは、現金挿入処理の終了によるリクエストにより、残高更新等の入金処理に移行し、コンピュータ処理の画面コンテンツをATM1に発行し、UOP6の顧客操作画面に、お待ち下さい画面を表示する。

【0100】

そして、ブラウザ120は、画面コンテンツに埋め込まれたアプレット名である同期制御Agent、メソッド名である印字/繰り出し/MSライト/排出準備により、同期制御Agentの印字/繰り出し/MSライト/排出準備メソッド(図7参照)を呼び出し、カードコントローラ172にMSライトコマンド、排出準備コマンドを、レシートコントローラ174に印字コマンド、排出準備コマンドを、ジャーナルコントローラ177に印字コマンドを発行する。

【0101】

これにより、カードユニット10は、カードの磁気ストライプに書き込みを行い、カードの排出を準備し、レシートプリンタ20は、レシートを印字し、排出の準備を行い、ジャーナルプリンタ70は、ジャーナルに印字を行う。各コントローラ172、174、177は、コマンド実行完了により、印字/繰り出し/MSライト/排出準備メソッドに完了を応答する。

【0102】

そして、印字/繰り出し/MSライト/排出準備メソッドは、取引状態(ここでは、排出準備完了)をPOST依頼し、POSTエージェント126が、Webサーバ100にリクエスト送信する。

【0103】

Webサーバ100のJSPは、媒体排出処理に移行し、媒体排出の画面コンテンツをATM1に発行する。ATM1では、ブラウザ120が、UOP6の顧客操作画面に、媒体排出画面を表示する。そして、ブラウザ120は、画面コンテンツに埋め込まれたアプレット名である同期制御Agent、メソッド名である媒体排出により、同期制御Agentの媒体排出メソッド(図7参照)を呼び出し、カードコントローラ172に放出コマンドを、レシートコントローラ174に放出コマンドを、図示しないサウンドコントローラに、ベル再生コマンドを発行する。

【0104】

これにより、カードユニット10は、カードを、レシートプリンタ20は、レシートを

放出し、カード、レシートの抜き取りを検出し、所定時間内に抜き取れないと、連続音を発する。各コントローラ 172, 174 は、コマンド実行完了により、媒体排出メソッドに完了を応答する。

【0105】

そして、媒体排出メソッドは、取引状態（ここでは、抜き取り完了）を POST 依頼し、POST エージェント 126 が、Web サーバ 100 にリクエスト送信する。

【0106】

次に、Web サーバ 100 の JSP は、ATM1 からの抜き取り完了によるリクエストを受け、取引終了処理に移行し、取引終了の画面コンテンツを ATM1 に発行する。ATM1 では、ブラウザ 120 により、UOP6 の顧客操作画面に、終了画面を表示する。ブラウザ 120 は、画面コンテンツに埋め込まれたアプレット名である同期制御 Agent、メソッド名である紙幣／硬貨収納により、同期制御 Agent の紙幣／硬貨メソッド（図 7 参照）を呼び出し、紙幣コントローラ 175 と硬貨コントローラ 176 に収納コマンドを発行する。

【0107】

これにより、紙幣ユニット、硬貨ユニット 50 は、計数紙幣、硬貨を内部スタックに収納する。各コントローラ 175, 176 は、コマンド実行完了により、紙幣／硬貨メソッドに完了を応答する。そして、紙幣／硬貨メソッドは、取引状態（ここでは、収納完了）を POST 依頼し、POST エージェント 126 が、Web サーバ 100 にリクエスト送信する。

【0108】

これにより、Web サーバ 100 は、前述の取引種別選択処理に戻り、同様な処理を繰り返す。

【0109】

このように、Web サーバ 100 の画面コンテンツに取引処理の各処理単位の Agent を埋め込み、処理単位の 1 つのメソッドを呼び出すことにより、複数の IO ユニットの動作制御でき、一般的な取引処理の流れで、異なる自動取引装置に共通に使用できる。

【0110】

このため、異なる機能、構成の自動取引装置を Web 制御する画面コンテンツの作成が容易となり、且つ並列制御が可能となり、高速の IO ユニットの制御を実現できる。これにより、利用者の自動取引装置の待ち時間を短縮でき、且つ稼働率を向上できる。

【0111】

この例では、カード入金処理で説明したが、出金処理、通帳を使用した入金処理、出金処理、記帳処理、残高照会等も同様である。

【0112】

[Agent によるリクエスト回数削減方法]

次に、Agent によるリクエスト回数の削減方法を説明する。図 14 乃至図 16 は、図 13 で説明した現金挿入画面の遷移を説明する図、図 17 は、図 4 の View Agent 128 の処理フロー図である。

【0113】

図 14 乃至図 16 の案内画面において、網掛け部分が、図 17 の View Agent 128 による画面更新可能エリア（即ち、Applet エリア）を示す。

【0114】

図 14 乃至図 16 に示すように、紙幣／硬貨挿入処理においては、紙幣／硬貨ユニット 50 の紙幣挿入、硬貨挿入、計数／鑑別、追加挿入、鑑別不良返却の動作に応じて、UOP6 の案内画面は、図 14 の初回入金画面から、紙幣挿入画面、図 15 の硬貨挿入画面、紙幣／硬貨挿入画面、追加挿入画面、図 16 の鑑別不良返却画面へ遷移する。

【0115】

即ち、図 14 に示すように、初回入金画面は、文言「紙幣と硬貨を入れてください。紙幣は、xxx 枚まで硬貨は、xxx 枚まで挿入できます。」という紙幣／硬貨の挿入指示

画面である。次に、紙幣／硬貨ユニット50に紙幣が挿入されると、図14に示すように、紙幣挿入画面に遷移し、文言は、「紙幣を計数しております。硬貨をお持ちでしたら投入して下さい。」の紙幣計数状態を示す。

【0116】

一方、紙幣／硬貨ユニット50に先に硬貨が挿入されると、図15に示すように、硬貨挿入画面に遷移し、文言は、「硬貨を計数しております。紙幣をお持ちでしたら投入して下さい。」の硬貨計数状態を示す。更に、紙幣／硬貨ユニット50に紙幣と硬貨とが挿入されると、図15に示すように、紙幣／硬貨挿入画面に遷移し、文言は、「紙幣と硬貨を計数しております。しばらくお待ち下さい。」の紙幣／硬貨計数状態を示す。

【0117】

更に、紙幣／硬貨ユニット50が、挿入紙幣／硬貨の鑑別を終了すると、図15に示すように、追加挿入画面に移行し、文言は、「金額をご確認のうえ、紙幣と硬貨をお入れ下さい。(金額が)よろしければ確認ボタンを押して下さい。」と「合計入金金額 x x x 円」の鑑別結果確認及び追加問い合わせ状態を示す。

【0118】

更に、紙幣／硬貨ユニット50が、挿入紙幣／硬貨の鑑別結果として鑑別不良を検出すると、図16のように、鑑別不良返却画面に移行し、文言は、「現金をお返しします。現金を抜き取り後、シャッタが閉じます。」と、「(鑑別不良でない現金の) 合計入金金額 x x x 円」の鑑別不良券返却及びそれ以外の合計金額を示す。

【0119】

このような画面の遷移を行うには、Web制御では、通常、画面更新毎に、Webサーバ100にリクエストを送信する必要があり、リクエスト回数が増大する。

【0120】

そこで、図14乃至図16に示すように、更新部分(網掛け部分)を、ViewAgent 128のアプレットエリアとし、図17に示すように、ViewAgent 128のアプレットを、初回入金画面から、紙幣挿入画面、硬貨挿入画面、紙幣／硬貨挿入画面、追加挿入画面、鑑別不良返却画面へ遷移するように構成する。そして、紙幣／硬貨Agentからの画面更新指示に応じて、必要なアプレットを呼び出し、画面の一部を変更する。

【0121】

このための制御方法として、図18及び図19で説明するApplet間通信による表示方法と、図20及び図21で説明するJavaScript呼び出しによる表示方法とを示す。尚、ここでは、紙幣又は硬貨の挿入フェーズのみを例として、示す。

【0122】

図18は、Applet間通信による表示制御動作の説明図、図19は、そのための画面コンテンツの記述内容の説明図である。図19に示すように、画面内で呼び出すAgentをアプレットタグで指定する。ここでは、紙幣又は硬貨の現金挿入のため、<APPLET CODE="U_agtSync__insertCash. ~">が記述される。又、ViewAgentをアプレットタグで指定する。即ち、<APPLET CODE="U_agtView" ~">が記述される。

【0123】

そして、ViewAgentのSCRIPTので、初期画面のパラメータとして、文言「紙幣／硬貨を挿入して下さい。」を記述する。又、ViewAgentの遷移画面の表示文言等(パラメータ)は、前述のViewAgentのアプレットタグのパラメータ定義で設定する。ここでは、<PARAM NAME>で設定され、文言は、「紙幣を鑑別しています。」を示す。

【0124】

図18により、この記述言語による動作を説明する。まず、図18の画面コンテンツのロード時に、ViewAgent 128は、画面コンテンツのパラメータ定義を取得する。次に、制御(現金)Agentのメソッド(insert Cash)が呼び出し、IOコントローラ175、176にコマンドを依頼する。IOコントローラ175、176は、紙幣／

硬貨ユニット50を制御し、その途中状態を、シャッタ開放、媒体(現金)検出、シャッタ閉鎖、鑑別開始時に、制御Agent124にレスポンスする。

【0125】

制御Agent124は、鑑別開始レスポンスを受けると、ViewAgent128に、画面更新を要求する。ViewAgent128は、これに応じて、取得したパラメータ定義の画面を作成し、UOP6に表示する。図19の記述では、文言定義が、「紙幣を鑑別しています。」であるため、図14に示したように、UOP6の画面は、初回入金画面から紙幣挿入画面に遷移する。

【0126】

以下、同様に、ViewAgent128による画面の一部更新が、制御Agent124の更新要求により行われる。そして、IOコントローラ175、176が、コマンド処理を完了すると、制御Agent124にコマンド完了を通知する。制御Agent124は、POSTAgent126にPOST依頼し、POSTAgent126が、POST処理(Webサーバ100へのリクエスト送信)する。

【0127】

この方法は、制御Agent124より対象となるイベントの受信タイミングで、Applet間通信により呼び出すものであり、スクリプティング量は小さくて済む利点がある。一方、画面更新タイミングが増える度に、アプレットタグのパラメータセットが増加し、制御Agent124とJSPの修正が必要となる。

【0128】

図20は、他の制御方法としてのJavaScript呼び出しによる表示制御動作の説明図、図21は、そのための画面コンテンツの記述内容の説明図である。図21に示すように、画面内で呼び出すAgentをアプレットタグで指定する。ここでは、紙幣又は硬貨の現金挿入のため、<APPLET CODE="U_agtSync__insertCash. ~">が記述される。又、ViewAgentをアプレットタグで指定する。即ち、<APPLET CODE="U_agtView" ~">が記述される。

【0129】

そして、SCRIPTで制御Agent124から呼び出される関数を定義する。即ち、"function insertCashFunc(eventKind, billVal, coinVal) {switch(eventKind) {case 1: 鑑別開始(紙幣) // ViewAgentの呼び出し、case 2: 鑑別開始(硬貨) // ViewAgentの呼び出し、~case n: 鑑別終了(紙幣/硬貨) // ViewAgentの呼び出し、で記述する。

【0130】

更に、SCRIPTので、初期画面のパラメータとして、文言「紙幣/硬貨を挿入して下さい。」等を記述する。図20により、この記述言語による動作を説明する。まず、画面コンテンツ122が、制御(現金)Agentのメソッド(insertCash)を呼び出し、IOコントローラ175、176にコマンドを依頼する。IOコントローラ175、176は、紙幣/硬貨ユニット50を制御し、その途中状態を、シャッタ開放、媒体(現金)検出、シャッタ閉鎖、鑑別開始時に、制御Agent124にレスポンスする。

【0131】

制御Agent124は、鑑別開始レスポンスを受けると、画面コンテンツ122のJavaScriptの関数(JSObject)を呼び出す。JavaScriptの関数内で、ViewAgent128をメソッド呼び出しする。文言は、呼び出された関数の引数(鑑別状態や鑑別枚数等)を判断して、選択される。これにより、ViewAgent128は、呼び出された関数の文言の画面を作成し、UOP6に表示する。図21の記述では、文言定義が、「紙幣を鑑別しています。」であるため、図20に示したように、UOP6の画面は、初回入金画面から紙幣挿入画面に遷移する。

【0132】

以下、同様に、ViewAgent128による画面の一部更新が、制御Agent124の更新要求により行われる。そして、IOコントローラ175、176が、コマンド処理を完了すると、制御Agent124にコマンド完了を通知する。制御Agent124は、POSTAgent126にPOST依頼し、POSTAgent126が、POST処理（Webサーバ100へのリクエスト送信）する。

【0133】

この方法は、図21で示したように、制御Agent124は、対称イベントの受信タイミングで、画面コンテンツ122のJavaScriptの関数（JSObject）を呼び出す。そして、JavaScriptの関数内で、ViewAgent128をメソッド呼び出しする。文言は、呼び出された関数の引数（鑑別状態や鑑別枚数等）を判断して、選択される。

【0134】

この方法は、画面更新タイミング時に、JavaScript関数を呼び出すため、ボタン制御等も連動でき、ViewAgentを、文言表示領域と金額表示領域の分割等の複数画面に組み込むことができる。一方、スクリプティング量は、図19の例に比し、増加する。

【0135】

〔リクエスト回数を削減した自動取引処理〕

次に、前述のAgentによるリクエスト回数削減方法を使用した自動取引処理を、図22乃至図24で説明する。図22乃至図24は、図14乃至図22で説明した紙幣／硬貨鑑別シーケンスを例に、案内画面、Script、ViewAgent、ControlAgent、紙幣コントローラ175、硬貨コントローラ176の処理シーケンスを示す。

【0136】

（S1）画面コンテンツのSCRIPT122が、制御（現金）Agent124の挿入メソッド（insert Cash）を呼び出し、制御Agent124は、ViewAgent128のメソッド（View1）を呼び出す。ViewAgent128は、UOP6に、「現金をお入れ下さい。」の文言の案内表示を行う。制御Agent124は、ViewAgent128からの応答を受け、紙幣コントローラ175、硬貨コントローラ176に受付／計数コマンドを発行する。紙幣コントローラ175、硬貨コントローラ176は、紙幣／硬貨ユニット50を制御し、その途中状態を、シャッタ開放、媒体（現金）検出、シャッタ閉鎖、鑑別開始時に、制御Agent124にレスポンスする。

【0137】

（S2）制御Agent124は、紙幣コントローラ175から鑑別開始レスポンスを受けると、ViewAgent128に、紙幣鑑別開始のメソッド呼び出しを行う。ViewAgent128は、これに応じて、UOP6に、文言定義が、「紙幣を計数しています。硬貨をお入れ下さい。」の案内表示を行う。即ち、図14に示したように、UOP6の画面は、初回入金画面から紙幣挿入画面に遷移する。

【0138】

（S3）更に、制御Agent124は、硬貨コントローラ176から鑑別開始レスポンスを受けると、ViewAgent128に、硬貨鑑別開始のメソッド呼び出しを行う。ViewAgent128は、これに応じて、UOP6に、文言定義が、「紙幣／硬貨を計数しています。」の案内表示を行う。即ち、図15に示したように、UOP6の画面は、紙幣挿入画面から紙幣／硬貨挿入画面に遷移する。

【0139】

（S4）制御Agent124は、紙幣及び硬貨コントローラ175、176から鑑別完了を受けると、鑑別枚数を保持する。取引状態を監視する取引制御Agent（図8参照）は、現金の受付待ち（例えば、硬貨無しの鑑別開始前）中かを判定する。受付待ち中であれば、取引制御Agentは、制御Agent124から、硬貨コントローラ176にキャンセルを依頼し、硬貨コントローラ176は、キャンセルを行い、キャンセル完了を応答する。そして、硬貨コントローラ176の制御を完了する。

【0140】

一方、受付待ち中でない場合や、硬貨コントローラ176の処理完了後に、コマンドの異常があったか（コントローラが異常か）を判定する。コマンド異常であれば、図24の「A」に進む。コマンド異常でない時は、入金枚数が限度枚数に一致するか、又は限度枚数をオーバーしているかを判定する。どちらの一方の場合には、同様に、図24の「A」に進む。更に、入金回数は、最大回数以上かを判定する。最大回数以上であれば、図24の「A」に進む。

【0141】

(S5) 最大回数以下であれば、図23に進み、制御Agent124は、ViewAgent128のメソッド (View1) を呼び出す。ViewAgent128は、UOP6のView1に、「追加の現金をお入れ下さい。」の文言の案内表示を行う。制御Agent124は、ViewAgent128からの応答を受け、ViewAgent128のメソッド (View2) を呼び出す。ViewAgent128は、UOP6のView2に、「入金金額 x x x x 円」の文言の案内表示を行う。即ち、図15の追加挿入画面に移行する。更に、制御Agent124は、ViewAgent128からの応答を受け、紙幣コントローラ175、硬貨コントローラ176に受付/計数コマンドを発行する。紙幣コントローラ175、硬貨コントローラ176は、紙幣/硬貨ユニット50を制御し、シャッタ開放し、これを制御Agent124にレスポンスする。

【0142】

(S6) その後、取引制御Agentは、終了キー（図15の確認キー）が押下されたかを判定し、押下されていない場合に、図22のステップS2に戻る。一方、終了キーが押下された場合には、Scriptから制御Agent124のキャンセルメソッドを呼び出す。制御Agent124から、紙幣及び硬貨コントローラ175、176にキャンセルを依頼し、紙幣及び硬貨コントローラ175、176は、キャンセルを行い、キャンセルコマンド完了を応答する。更に、制御Agent124から、紙幣/硬貨コントローラ175、176にシャッタクローズ用排出を依頼し、紙幣/硬貨コントローラ175、176は、シャッタクローズを行い、コマンド完了を応答する。制御Agent124は、POSTAgent126にPOST依頼し、POSTAgent126が、POST処理 (Webサーバ100へのリクエスト送信) する。

【0143】

(S7) 一方、図24の「A」に進むと、制御Agent124は、ViewAgent128のメソッド (View1) を呼び出す。ViewAgent128は、UOP6のView1に、「シャッタを閉めます。」の文言の案内表示を行う。制御Agent124は、ViewAgent128からの応答を受けた後、取引制御Agentは、返却現金があるかを判定する。返却現金がある場合には、制御Agent124に通知し、制御Agent124は、紙幣/硬貨コントローラ175、176に現金排出を依頼し、紙幣/硬貨コントローラ175、176は、シャッタ開放を行い、現金抜き取りを検出すると、シャッタを閉じ、コマンド完了を応答する。

【0144】

(S8) 取引制御Agentは、返却現金がない場合や、返却現金の排出を完了した場合には、シャッタが開放しているかを確認する。シャッタが開放している場合には、制御Agent124に通知し、制御Agent124は、紙幣/硬貨コントローラ175、176に現金排出を依頼し、紙幣/硬貨コントローラ175、176は、シャッタ開放を行い、現金抜き取りを検出すると、シャッタを閉じ、コマンド完了を応答する。シャッタが開放していない場合や、排出処理が完了している場合には、制御Agent124は、POSTAgent126にPOST依頼し、POSTAgent126が、POST処理 (Webサーバ100へのリクエスト送信) する。

【0145】

このように、ステップS2乃至S6で示したように、追加入金を行う場合に、1回の入金毎に、Webサーバ100と通信を行うのではなく、何回かの追加入金が全て完了した

後、まとめて通信する（POST処理する）ので、Agentにて、Webサーバに判断を仰ぐ必要がない場合には、Agentにて、処理を判断して、通信回数を削減できる。

【0146】

同様に、画面の更新を全てWebサーバからダウンロードするのではなく、Agentで、画面の一部を更新するため、Webサーバとの通信を行わずに、画面の更新が可能となる。ここでは、入金画面の文言更新を行い、且つ追加入金時の文言更新できる。

【0147】

このため、Webサーバ100へのリクエスト回数が減少し、通信回数及び通信時間を減少でき、自動取引装置の取引性能（速度）を向上でき、且つWebサーバの負荷を軽減できる。

【0148】

同様に、図9で説明したPOST処理のタイマーにより、Webサーバとの通信が一定時間ないと、処理を中断する。これにより、Webサーバに通信の問い合わせを行わずに済み、通信回数の低減が可能となり、且つWebサーバとの通信切断時にも、クライアント（ATM）側で、処理を中断できる。

【0149】

【他の実施の形態】

次に、前述のAgent、メソッドの他の実施の形態を説明する。図25は、本発明の他の実施の形態のエージェントの説明図である。図25に示すように、この例では、同期Agentと、メソッド（プログラム）を別体とし、同期Agentは、メソッドとして、初期化、メカリセット、紙幣／硬貨挿入、媒体同時排出、印字／繰り出し／MSライト／排出準備、入金返却、収納、強制排出／取込み、ユニット情報取得／取引状態設定／二面表示制御、入出金準備、強制補充、ジャムリセット、カード／通帳挿入を有する。

【0150】

各メソッドは、図25の黒丸で示すIOコントローラ（即ち、IOユニット）にコマンドを発行し、コマンド実行結果の応答を受ける。例えば、同期制御エージェントの初期化メソッドは、紙幣コントローラ175、硬貨コントローラ176、通帳コントローラ171、カードコントローラ172、レシートコントローラ174、ジャーナルコントローラ177、取引制御コントローラ178、テンキーコントローラ173（図5参照）に、初期化コマンドを発行し、各コントローラによる初期化処理を行わせ、各コントローラから初期化処理結果を応答として、受信する。

【0151】

同様に、同期制御エージェントのメカリセットメソッドは、紙幣コントローラ175、硬貨コントローラ176、通帳コントローラ171、カードコントローラ172、レシートコントローラ174、ジャーナルコントローラ177（図5参照）に、メカリセットコマンドを発行し、各コントローラによるメカリセット処理を行わせ、各コントローラからメカリセット処理結果を応答として、受信する。

【0152】

以下、同様に、同期制御エージェントの紙幣／硬貨挿入メソッドは、紙幣コントローラ175、硬貨コントローラ176（図5参照）に、挿入コマンドを発行し、各コントローラによる挿入処理を行わせ、各コントローラから挿入処理結果を応答として、受信する。

【0153】

図26は、本発明の他の実施の形態におけるWebサーバ100が、送信する画面コンテンツの説明図である。

【0154】

図26に示すように、HTML（ページ記述言語）で記述された画面コンテンツ内で、画面内で呼び出すAgent名を指定する。ここでは、一例として、アプレットタグ<APPLET CODE="U__agtSync_class"により、同期制御（Sync）エージェントを呼び出すことを示す。

【0155】

更に、`<SCRIPT Language="JavaScript">`の記述により、Java Scriptで、同期制御の初期化メソッドとして呼び出しを指定する。即ち、`ret=document. U_agtSync__initial`の記述で、同期制御の初期化メソッドの呼び出しを指定する。尚、ここでは、画面表示コンテンツの詳細を省略してある。

【0156】

この画面コンテンツが、ブラウザ120にダウンロードされると、アプレットタグにより、同期制御の初期化エージェントが指定され、且つSCRIPTの呼び出しメソッド名により、同期制御の初期化メソッドが呼び出される。

【0157】

前述のように、呼び出された初期化メソッドは、紙幣コントローラ175、硬貨コントローラ176、通帳コントローラ171、カードコントローラ172、レシートコントローラ174、ジャーナルコントローラ177、取引制御コントローラ178、テンキーコントローラ173（図5参照）に、初期化コマンドを発行し、各コントローラによる初期化処理を行わせ、各コントローラから初期化処理結果を応答として、受信する。

【0158】

この場合にも、図12に示したように、呼び出された初期化メソッドは、紙幣コントローラ175、硬貨コントローラ176、通帳コントローラ171、カードコントローラ172、レシートコントローラ174、ジャーナルコントローラ177、取引制御コントローラ178、テンキーコントローラ173（図5参照）に、初期化コマンドを連続的に発行し、これらコントローラからの応答を、処理完了したものから順次受ける。

【0159】

このため、IOユニットを並列的に制御でき、複数のIOユニットを制御しても、短時間で制御できる。これに比し、図29で示したように、従来のScriptで、IOユニット単位のメソッドを指定する方法では、Script毎に、コマンド発行、応答受信を順次行う順次制御のため、WebによるIO制御に時間がかかる。

【0160】

更に、前述のAgentは、処理単位名が付された処理単位で設けられるため、この処理単位の内容を若干変更するだけで、各種の機能のATMに、同一のアプレットタグ、メソッド名で、対応できる。この例でも、前述のメソッド呼び出しSCRIPTのメソッド名の括弧内に設けられた`initial(input Param)`の変更により実現している。即ち、図11に示したように、`input Param`は、各IOコントローラ（紙幣、硬貨、通帳、カード、レシート、ジャーナル、取引制御、テンキー）に、入力フラグを設定する8ビットの情報で構成されている。

【0161】

入力フラグは、“1”がセットされたIOコントローラへのメソッドの入力を許可することを示す。呼び出されたメソッドは、この入力フラグを参照して、コマンドを発行するIOコントローラを決定する。このため、入力フラグをWebサーバ100で操作することにより、異なる機能のATMでも、同一のアプレットタグ、メソッド名で、同一の処理を実行させることができる。

【0162】

例えば、硬貨を扱わない取引装置では、硬貨の入力フラグを“0”にセットしておけば、硬貨コントローラへのコマンド発行を防止できる。同様に、通帳を扱わない取引装置では、通帳の入力フラグを“0”にセットしておけば、通帳コントローラへのコマンド発行を防止できる。

【0163】

図7の実施の形態では、画面コンテンツとともに送信するアプレット（エージェント）が、図25の例に比し、比較的小容量のプログラムに分割できるため、送信コンテンツのデータ量を少なくでき、通信時間を短縮できる。

【0164】

一方、図25のエージェントの例では、同期制御のアプレットが、多数のメソッドを搭

載しているため、比較的大容量のプログラムであるが、ATM1で、アプレットをキャッシュすることにより、次回からこのアプレットの送信を省略でき、次回からの送信コンテンツのデータ量を少なくでき、通信時間を短縮できる。

【0165】

前述の実施の形態では、図1のような自動取引装置として、自動入出金機を例に、説明したが、出金機、自動現金貸出機、自動発行機等の他の装置に適用できる。又、ネットワークをインターネットで説明したが、他のネットワークに適用でき、且つサーバは、SCRIPTは、Java SCRIPTのみならず、他のSCRIPTのものにも適用できる。

【0166】

又、画面の更新を、入金処理で説明したが、暗証入力画面において、暗証入力時に訂正ボタンを押した時の再入力案内画面への更新や、通帳、カードの再挿入案内画面への更新等にも適用できる。更に、共通APIによるカスタマイズ化したミドルウェア制御を例に説明したが、これを使用せず、ミドルウェア制御するものにも、適用できる。

【0167】

以上、本発明を実施の形態により説明したが、本発明の趣旨の範囲内において、本発明は、種々の変形が可能であり、本発明の範囲からこれらを排除するものではない。

【0168】

(付記1) 利用者の操作に応じて、Webサーバと通信し、案内表示と取引動作を行う自動取引装置において、前記案内表示を行う表示ユニットと、前記取引動作を行うための複数のIOユニットと、前記Webサーバからの画面コンテンツに応じて、前記表示装置の画面の案内表示を制御し、且つ前記画面コンテンツに埋め込まれたオブジェクトに応じて、前記複数のIOユニットを制御する制御ユニットとを有し、前記制御ユニットは、前記オブジェクトのスクリプトにより、前記IOユニットをシーケンス制御するとともに、前記表示ユニットの表示文言を前記シーケンス制御に応じて、更新することを特徴とする自動取引装置。

【0169】

(付記2) 前記制御ユニットは、前記Webサーバからの画面コンテンツのアプレットタグを解釈し、前記案内表示を行うとともに、前記画面コンテンツに埋め込まれた前記オブジェクトのアプレットタグを解釈し、前記IOユニットを制御する処理単位のメソッドを呼び出すブラウザを有し、前記ブラウザから前記IOユニットをシーケンス制御し、且つ前記シーケンス制御に応じて、前記表示ユニットの表示文言を更新することを特徴とする付記1の自動取引装置。

【0170】

(付記3) 前記制御ユニットは、前記オブジェクトのアプレットと前記画面のアプレットとの通信を行い、前記表示ユニットの表示文言を更新することを特徴とする付記2の自動取引装置。

【0171】

(付記4) 前記制御ユニットは、前記オブジェクトのアプレットにより、前記スクリプトの関数を呼び出し、前記表示ユニットの表示文言を更新することを特徴とする付記2の自動取引装置。

【0172】

(付記5) 前記制御ユニットは、前記画面コンテンツに埋め込まれたオブジェクトのスクリプトに応じて、前記複数のIOユニットを同期制御する処理単位のメソッドを呼び出し、前記複数のIOユニットを同期制御することを特徴とする付記1の自動取引装置。

【0173】

(付記6) 前記IOユニットは、少なくとも現金を受け入れ、受け入れた現金を鑑別して、入金する現金ユニットを含み、前記制御ユニットは、前記現金ユニットの動作に応じて、前記画面コンテンツのスクリプトを解釈し、前記表示ユニットの表示文言を更新することを特徴とする付記1の自動取引装置。

【0174】

(付記7) 前記制御ユニットは、初回入金と追加入金とを実行した後、前記Webサーバにリクエストを送信することを特徴とする付記6の自動取引装置。

【0175】

(付記8) 利用者の操作に応じて、Webサーバと通信し、案内表示と取引動作を行う自動取引装置の制御方法において、前記Webサーバからの画面コンテンツに応じて、表示装置の画面の案内表示を行うステップと、前記画面コンテンツに埋め込まれたオブジェクトに応じて、IOユニットをシーケンス制御するステップと、前記オブジェクトにより、前記IOユニットのシーケンス制御に応じて、前記表示ユニットの表示文言を更新するステップとを有することを特徴とする自動取引装置の制御方法。

【0176】

(付記9) 案内表示ステップは、ブラウザにより、前記Webサーバからの画面コンテンツのアプリレットタグを解釈し、前記案内表示を行うステップからなり、前記制御ステップは、前記ブラウザにより、前記画面コンテンツに埋め込まれた前記オブジェクトのアプリレットタグを解釈し、前記IOユニットを制御する処理単位のメソッドを呼び出すステップからなり、前記更新ステップは、前記ブラウザから前記IOユニットをシーケンス制御したことに応じて、前記表示ユニットの表示文言を更新するステップからなることを特徴とする付記8の自動取引装置の制御方法。

【0177】

(付記10) 前記更新ステップは、前記オブジェクトのアプリレットと前記画面のアプリレットとの通信を行い、前記表示ユニットの表示文言を更新するステップからなることを特徴とする付記9の自動取引装置の制御方法。

【0178】

(付記11) 前記更新ステップは、前記オブジェクトのアプリレットにより、前記スクリプトの関数を呼び出し、前記表示ユニットの表示文言を更新するステップからなることを特徴とする付記9の自動取引装置の制御方法。

【0179】

(付記12) 前記制御ステップは、前記画面コンテンツに埋め込まれたオブジェクトのスクリプトに応じて、前記複数のIOユニットを同期制御する処理単位のメソッドを呼び出し、前記複数のIOユニットを同期制御するステップからなることを特徴とする付記8の自動取引装置の制御方法。

【0180】

(付記13) 前記更新ステップは、前記IOユニットとして設けられた、少なくとも現金を受け入れ、受け入れた現金を鑑別して、入金する現金ユニットの動作に応じて、前記画面コンテンツのスクリプトを解釈し、前記表示ユニットの表示文言を更新するステップからなることを特徴とする付記8の自動取引装置の制御方法。

【0181】

(付記14) 前記現金ユニットが、初回入金と追加入金とを実行した後、前記Webサーバにリクエストを送信するステップを更に有することを特徴とする付記13の自動取引装置の制御方法。

【0182】

(付記15) Webサーバと、利用者の操作に応じて、前記Webサーバと通信し、案内表示と取引動作を行う自動取引装置とを有し、前記自動取引装置は、前記案内表示を行う表示ユニットと、前記取引動作を行うための複数のIOユニットと、前記Webサーバからの画面コンテンツに応じて、前記表示装置の画面の案内表示を制御し、且つ前記画面コンテンツに埋め込まれたオブジェクトに応じて、前記複数のIOユニットを制御する制御ユニットとを有し、前記制御ユニットは、前記オブジェクトのスクリプトにより、前記IOユニットをシーケンス制御するとともに、前記表示ユニットの表示文言を前記シーケンス制御に応じて、更新することを特徴とする自動取引システム。

【0183】

(付記 16) 前記自動取引装置の前記制御ユニットは、前記 Web サーバからの画面コンテンツのアプレットタグを解釈し、前記案内表示を行うとともに、前記画面コンテンツに埋め込まれた前記オブジェクトのアプレットタグを解釈し、前記 I/O ユニットの制御する処理単位のメソッドを呼び出すブラウザを有し、前記ブラウザから前記 I/O ユニットのシーケンス制御し、且つ前記シーケンス制御に応じて、前記表示ユニットの表示文言を更新することを特徴とする付記 15 の自動取引システム。

【0184】

(付記 17) 前記制御ユニットは、前記オブジェクトのアプレットと前記画面のアプレットとの通信を行い、前記表示ユニットの表示文言を更新することを特徴とする付記 16 の自動取引システム。

【0185】

(付記 18) 前記制御ユニットは、前記オブジェクトのアプレットにより、前記スクリプトの関数を呼び出し、前記表示ユニットの表示文言を更新することを特徴とする付記 16 の自動取引システム。

【0186】

(付記 19) 前記制御ユニットは、前記画面コンテンツに埋め込まれたオブジェクトのスクリプトに応じて、前記複数の I/O ユニットの同期制御する処理単位のメソッドを呼び出し、前記複数の I/O ユニットの同期制御することを特徴とする付記 15 の自動取引システム。

【0187】

(付記 20) 前記 I/O ユニットの、少なくとも現金を受け入れ、受け入れた現金を鑑別して、入金する現金ユニットを含み、前記制御ユニットは、前記現金ユニットの動作に応じて、前記画面コンテンツのスクリプトを解釈し、前記表示ユニットの表示文言を更新することを特徴とする付記 15 の自動取引システム。

【0188】

(付記 21) 前記制御ユニットは、初回入金と追加入金とを実行した後、前記 Web サーバにリクエストを送信することを特徴とする付記 20 の自動取引システム。

【産業上の利用可能性】

【0189】

このように、Web サーバの画面コンテンツに取引処理の各処理単位のアプレットとスクリプトを埋め込み、I/O ユニットの動作制御と画面更新を行うため、Web サーバとの通信回数を減少でき、自動取引装置の性能向上ができ、Web 制御の自動取引装置の普及に寄与する。

【図面の簡単な説明】

【0190】

- 【図 1】 本発明の一実施の形態の自動取引装置の外観図である。
- 【図 2】 図 1 の自動取引装置のブロック図である。
- 【図 3】 本発明の一実施の形態の自動取引システムのシステム構成図である。
- 【図 4】 図 3 のブラウザ及び画面コンテンツの説明図である。
- 【図 5】 図 3 及び図 4 の ATM ミドルウェアのブロック図である。
- 【図 6】 図 3 及び図 5 の共通インタフェースの取引コマンドの説明図である。
- 【図 7】 図 3 及び図 4 のエージェントの説明図である。
- 【図 8】 図 3 及び図 4 の他のエージェントの説明図である。
- 【図 9】 図 3 及び図 4 の画面コンテンツの説明図である。
- 【図 10】 図 3 及び図 4 のエージェントによる I/O 制御動作の説明図である。
- 【図 11】 図 9 の画面コンテンツの入力パラメータの説明図である。
- 【図 12】 図 10 のエージェントのメソッドの動作説明図である。
- 【図 13】 本発明の一実施の形態の入金処理の説明図である。
- 【図 14】 本発明の一実施の形態の第 1 の案内画面更新動作説明図である。
- 【図 15】 本発明の一実施の形態の第 2 の案内画面更新動作説明図である。

【図 16】本発明の一実施の形態の第 3 の案内画面更新動作説明図である。

【図 17】本発明の一実施の形態の View Agent の画面更新処理フロー図である。

【図 18】本発明の一実施の形態の I/O 制御と画面更新処理のシーケンス図である。

【図 19】図 18 の実施の形態の画面コンテンツの記述例の説明図である。

【図 20】本発明の他の実施の形態の I/O 制御と画面更新処理のシーケンス図である。

【図 21】図 20 の実施の形態の画面コンテンツの記述例の説明図である。

【図 22】本発明の一実施の形態の紙幣／硬貨入金処理フロー図（その 1）である。

【図 23】本発明の一実施の形態の紙幣／硬貨入金処理フロー図（その 2）である。

【図 24】本発明の一実施の形態の紙幣／硬貨入金処理フロー図（その 3）である。

【図 25】本発明の他の実施の形態のエージェントの説明図である。

【図 26】本発明の他の実施の形態の画面コンテンツの説明図である。

【図 27】従来の Web 制御による自動取引システムの説明図である。

【図 28】従来の Web 制御の画面コンテンツの説明図である。

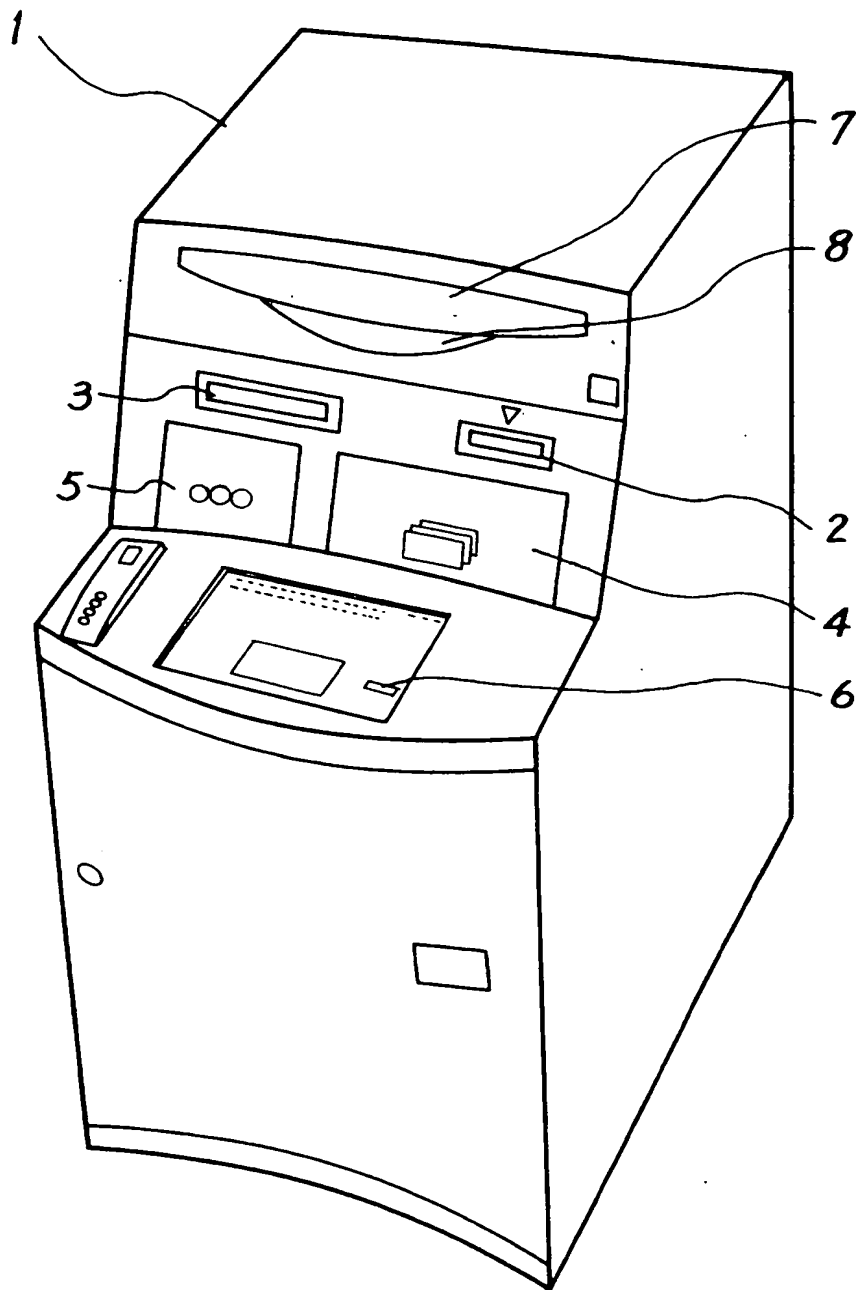
【図 29】従来の画面コンテンツのスクリプトによる I/O 制御動作のシーケンス図である。

【符号の説明】

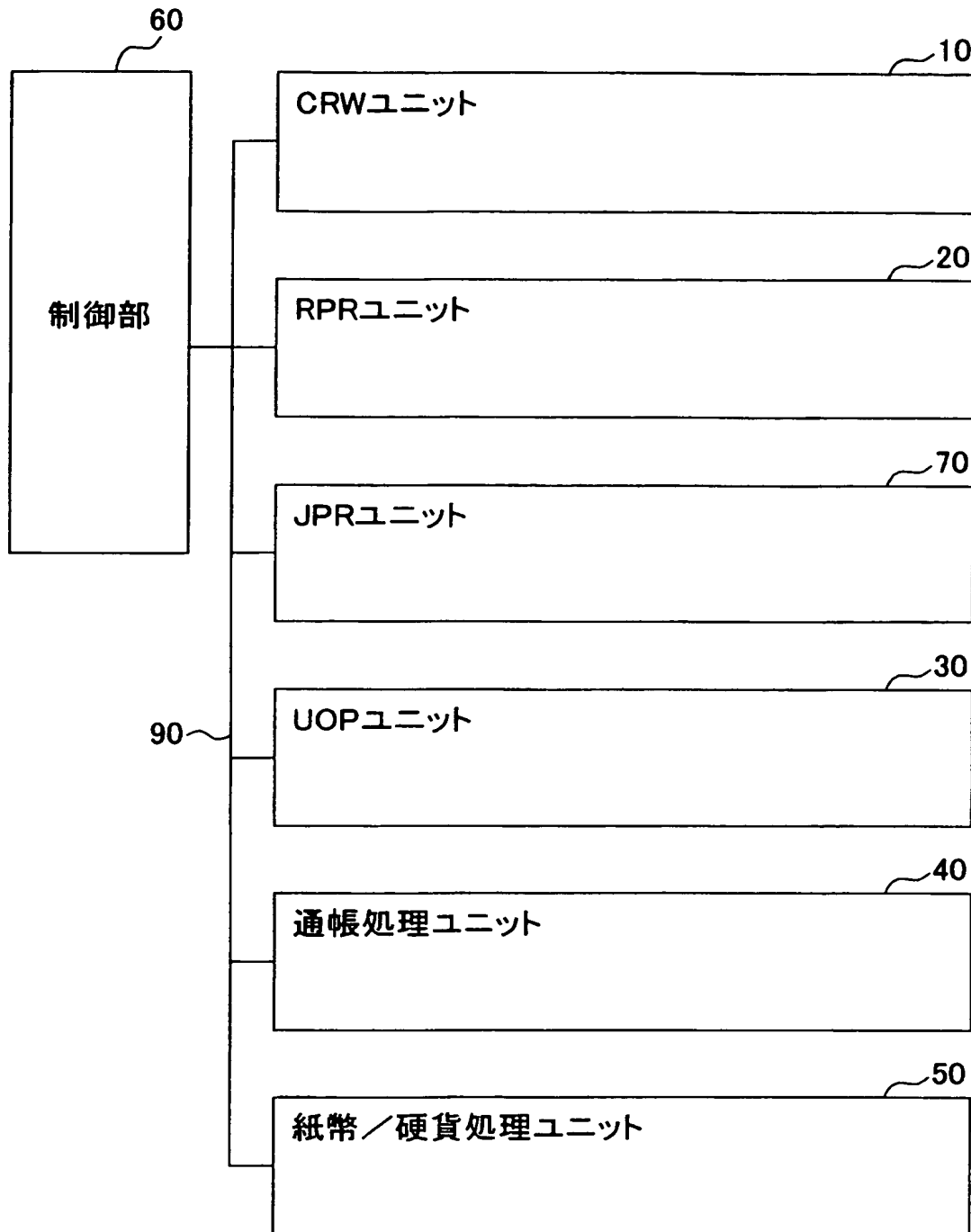
【0191】

- 1 自動取引装置
- 2 カード出入口
- 3 通帳出入口
- 4 紙幣出入口
- 5 硬貨出入口
- 6 UOP
- 10 磁気カードリーダユニット
- 20 レシートプリンタユニット
- 30 UOP ユニット
- 40 通帳処理ユニット
- 50 紙幣／硬貨処理ユニット
- 60 制御ユニット
- 70 ジャーナルプリンタユニット
- 100 Web サーバ
- 110 ネットワーク
- 120 ブラウザ
- 122 画面コンテンツ
- 124、126、128 エージェント
- 130 ATM ミドルウェア
- 140 カーネル
- 150 デバイスドライバ
- 160 パラメータファイル
- 170 I/O コントロール層
- 180 既存ミドルウェア層

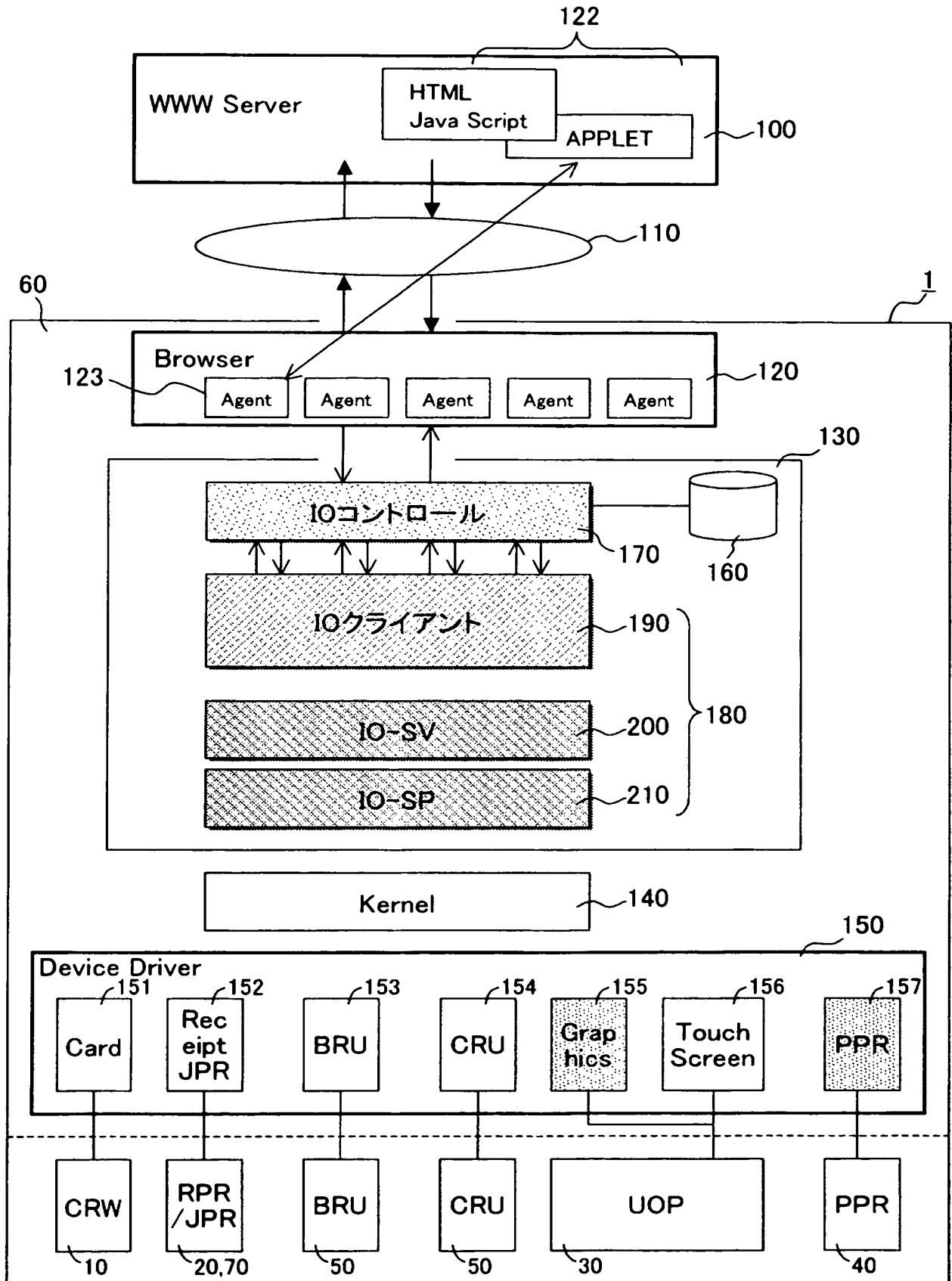
【書類名】図面
【図1】



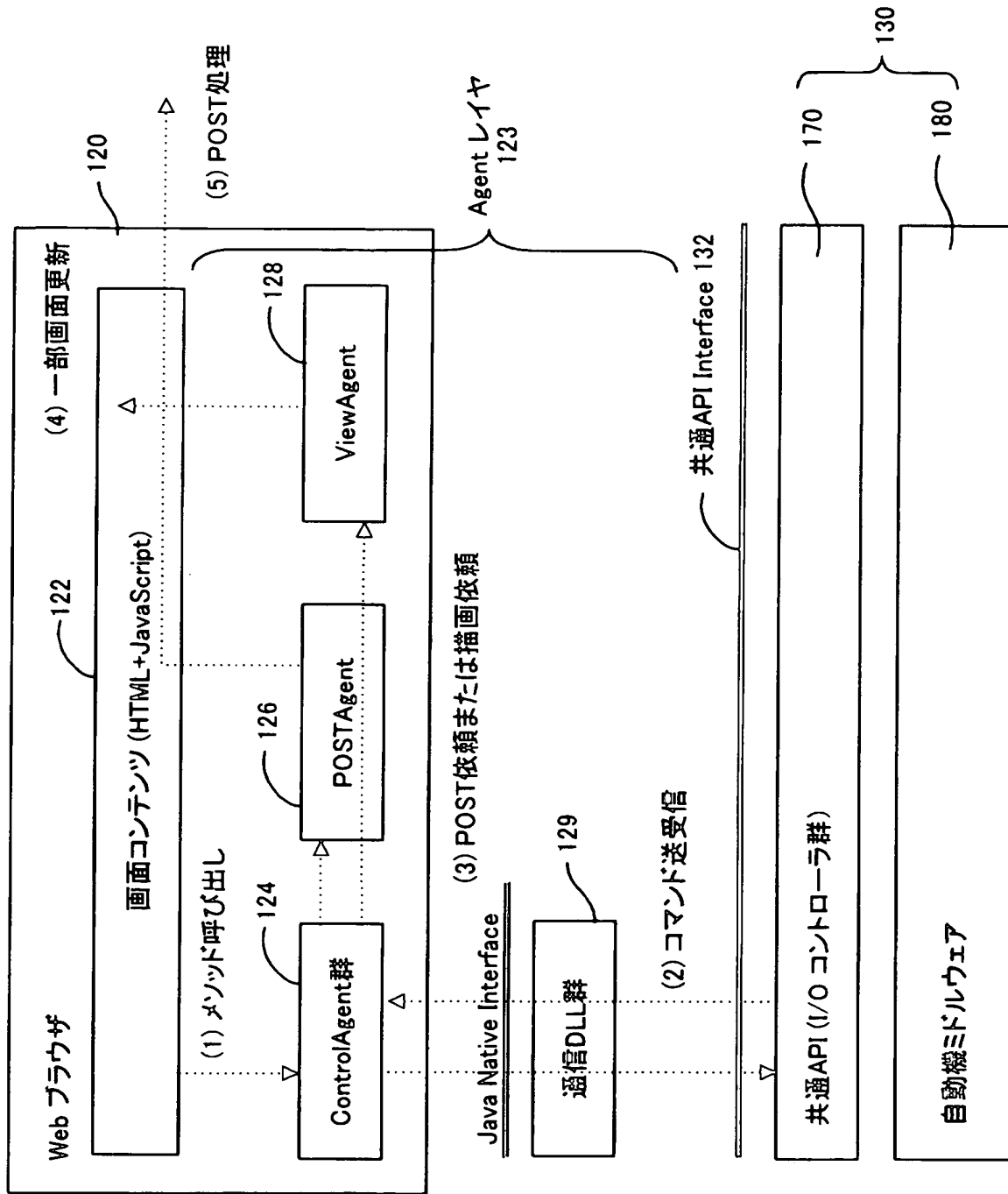
【図 2】



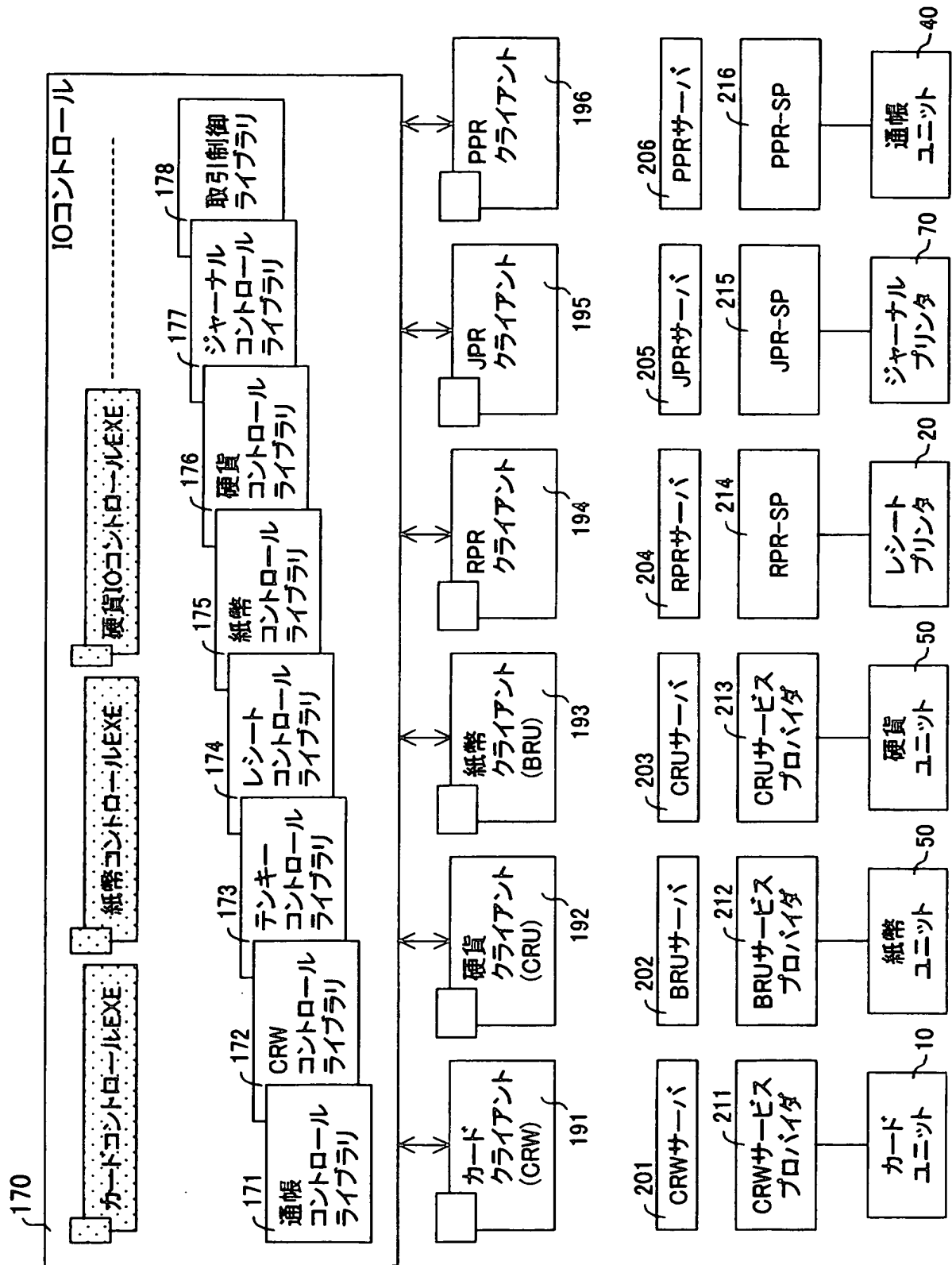
【図 3】



【図 4】



【図 5】



【図 6】

コマンド種別	コマンド
CRW コマンド	カード挿入
	カード排出
RPR コマンド	印字
	放出
PPR コマンド	通帳挿入
	印字
	MS書き込み
	通帳排出
	オートターンページ
BRU コマンド (CRU コマンド)	初期化
	受付／計数
	収納
	入金返却
	繰出
	放出
	取込
	搬送路チェック
	ジャムリセット

【図 7】

Agent		(method)	対象IOコントロール								
			なし	紙幣	硬貨	通帳	カード	レシート	ジャーナル	取引	テンキー
同期制御	初期化		●	●	●	●	●	●	●	●	
同期制御	メモリセット		●	●	●	●	●	●	●		
同期制御	紙幣／硬貨挿入		●	●	●	●	●	●	●		
同期制御	媒体同時排出		●	●	●	●	●				
同期制御	印字／繰出／MSライト／排出準備		●	●	●	●	●	●			
同期制御	入金返却		●	●	●	●	●				
同期制御	収納		●	●	●	●	●				
同期制御	強制排出／取込み		●	●	●	●	●				
同期制御	ユニット情報取得／取引状態設定／二面表示制御		●	●	●	●	●	●	●	●	
同期制御	キャンセル										
同期制御	入金準備		●	●	●	●	●				
同期制御	強制補充		●	●	●	●	●				
同期制御	ジャムリセット		●	●	●	●	●				
同期制御	カード／通帳挿入					●					
POST	POST処理	●									
	POSTデータ保持	●									
テキスト表示	フォント設定	●									
	テキスト表示	●									
	テキスト消去	●									

【図 8】

Agent	method	対象IOコントロール								
		なし	紙幣	硬貨	通帳	カード	レシート	ジャーナル	取引	テンキー
紙幣制御	受付／計数		○							
	収納		○							
	入金返却		○							
	放出		○							
	キャンセル		○							
通帳制御	行セット／ページマークリード				○					
	MSリード				○					
	オートターンページ				○					
	ページチェックオートターン				○					
	通帳構成情報設定				○					
カード制御	カード挿入					○				
	キャンセル					○				
	振込みカード印字					○				
	振込みカード発行					○				
	排出準備					○				
レシート制御	オーバーレイ登録						○			
	排出準備						○			
取引制御	取引情報設定								○	
	装置状態監視								○	
	装置状態取得								○	
	運用情報設定								○	
	キャンセル								○	
テンキー	テンキー入力開始									○
	テンキー入力終了									○

【図 9】

```

<HTML>
<HEAD>
<TITLE>WebATM Sample</TITLE>
</HEAD>

<BODY>

// 画面内で呼び出す Agent を指定する.
<APPLET CODE="U_agtSync_initial.class" codebase = "http://webatm:8080/webatm"
        WIDTH="0" HEIGHT="0" NAME="U_agtSync_initial" >
</APPLET>

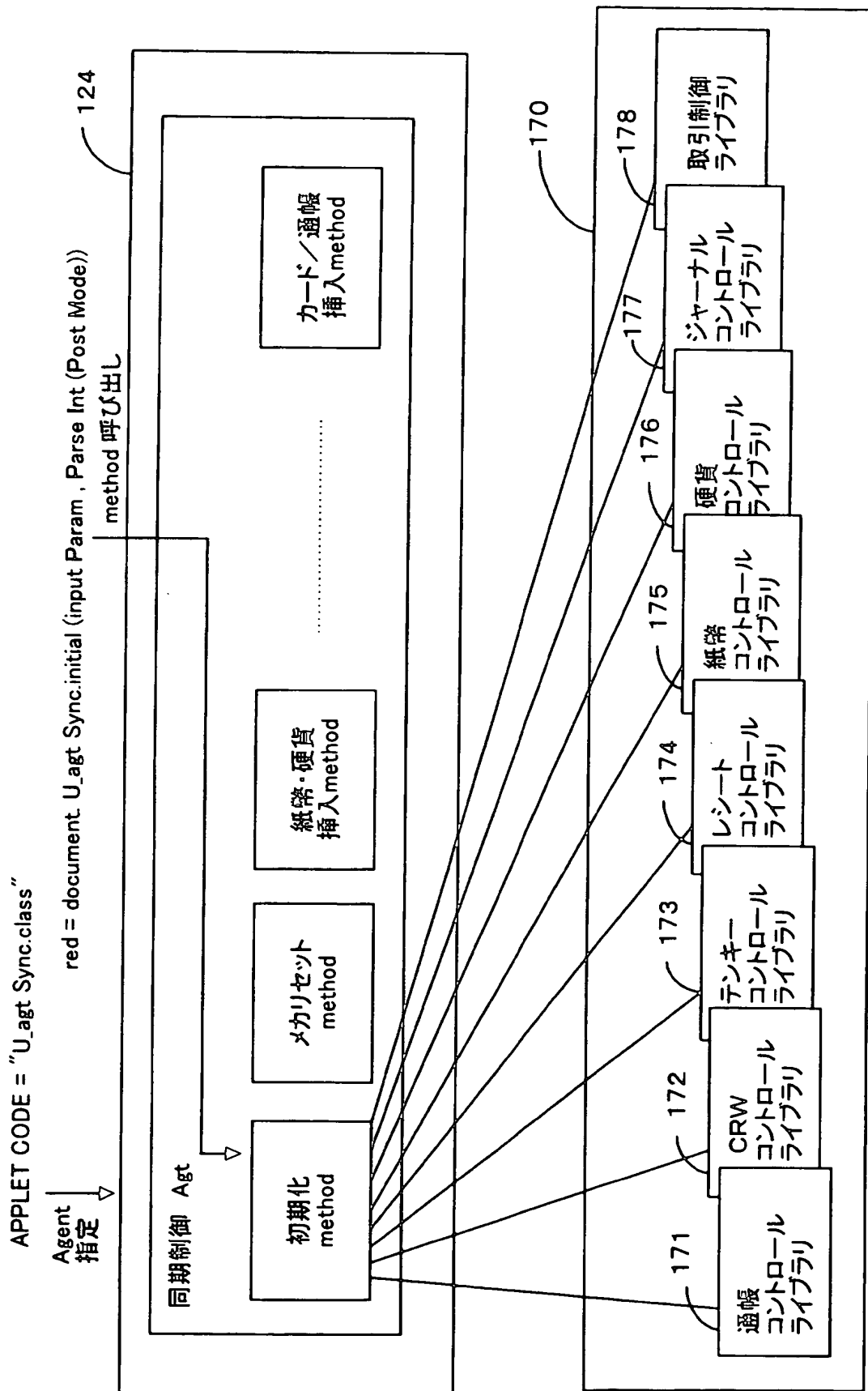
// POSTAgent が JavaScript の関数を呼び出せるように MAYSCRIPT 定義を行う.
<APPLET CODE="U_agtPost.class" codebase = "http://webatm:8080/webatm"
        WIDTH="0" HEIGHT="0" NAME="U_agtPost" MAYSCRIPT>

<SCRIPT language="javascript">
<!--
// POSTAgent から呼び出される関数を定義する. (必須)
function postFunc(postData){document.sys_form.ioResponse.value = postData;
        window.document.sys_form.submit0;
}
// 同期 Agent の初期化メソッドの呼び出し
ret = document.U_agtSync_initial.initial (inputParam, parseInt(postMode), parseInt(maxTimer));
if (ret < 0) {
        // 異常通知処理
        ~省略~
}

// -->
</SCRIPT>
</BODY>
</HTML>

```

【図10】



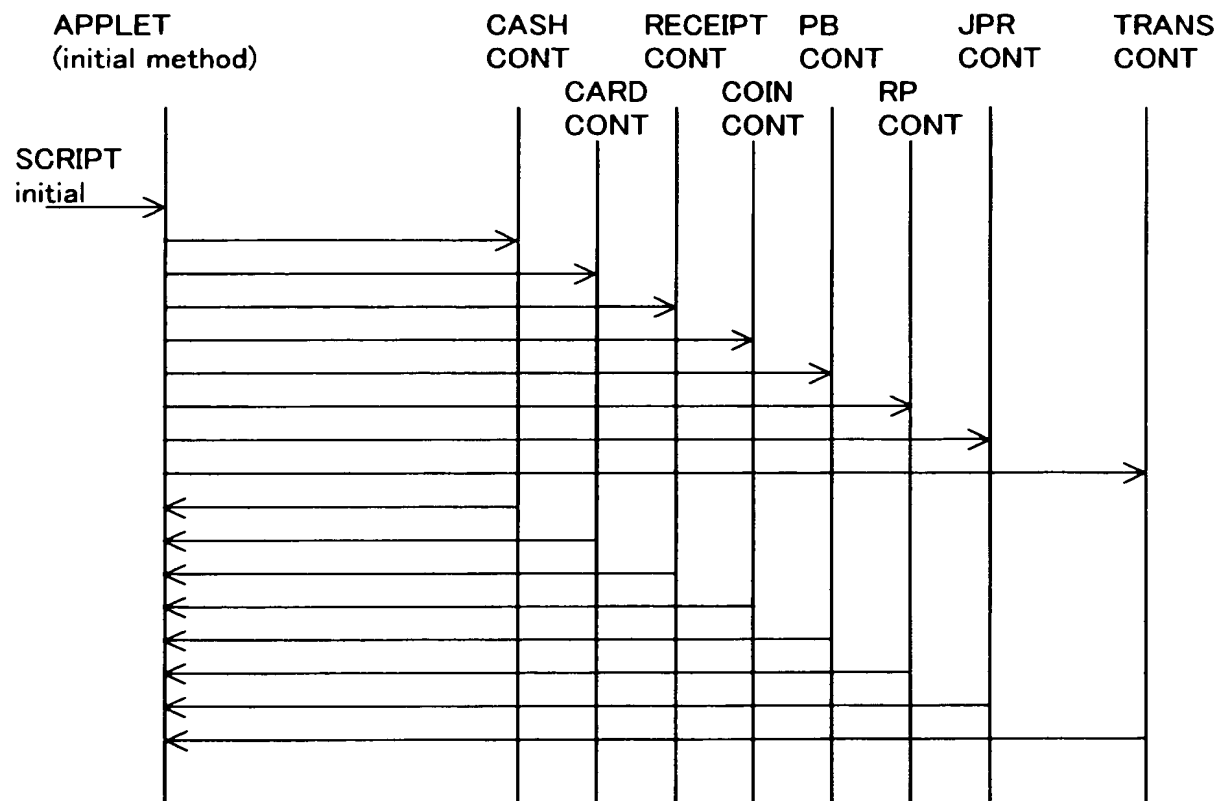
【図 1 1】

Input Param

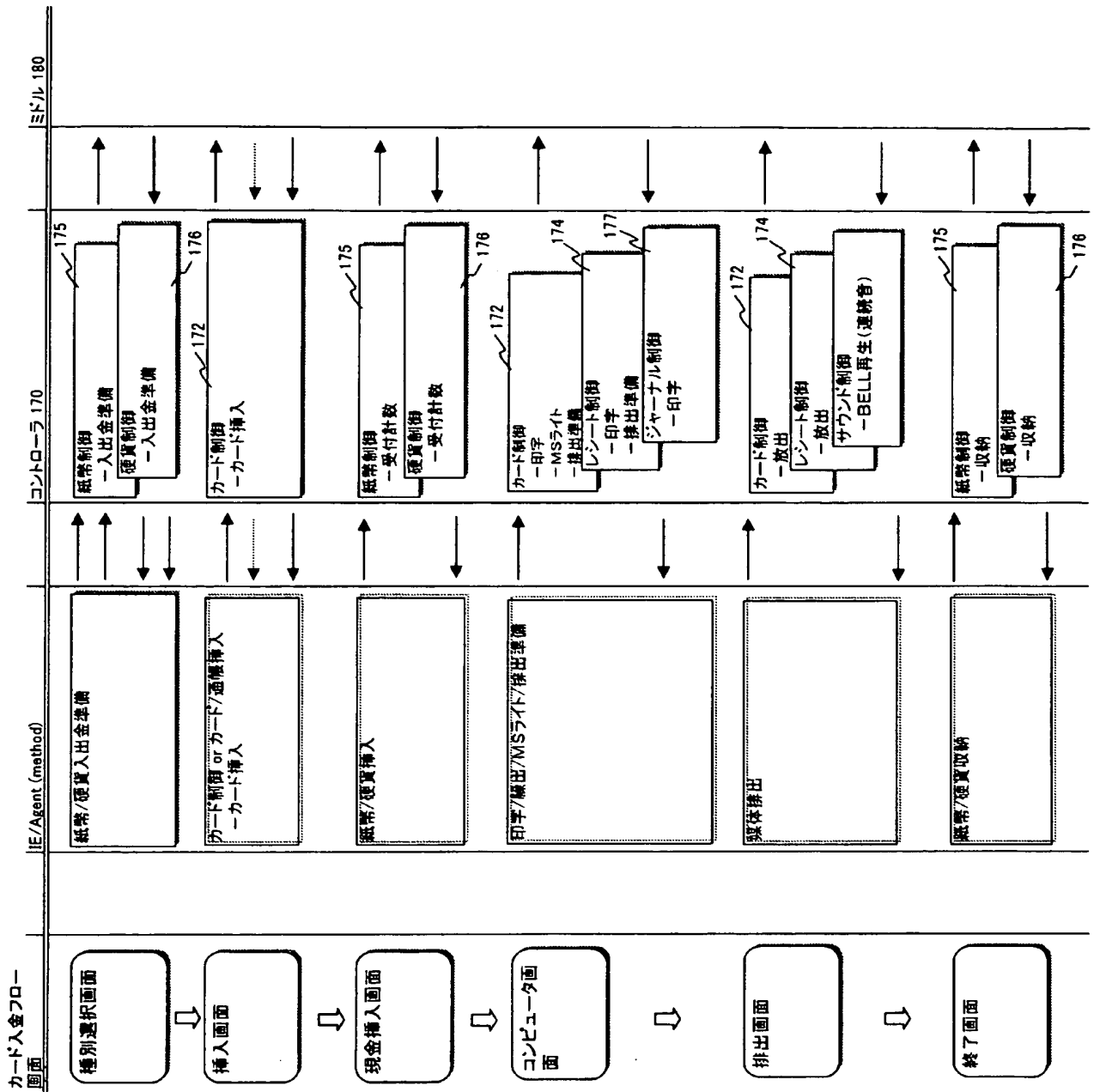
1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

紙幣 硬貨 通帳 カード レシート ジャーナル 取引 テンキー

【図 1 2】

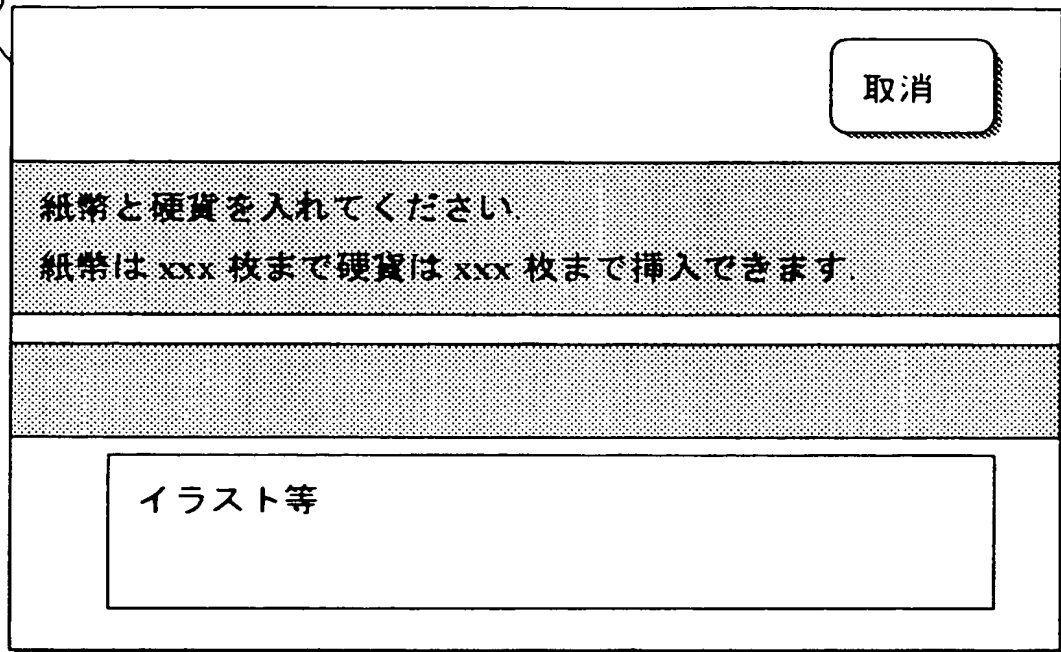


【図 13】



【図 14】

①初回入金画面



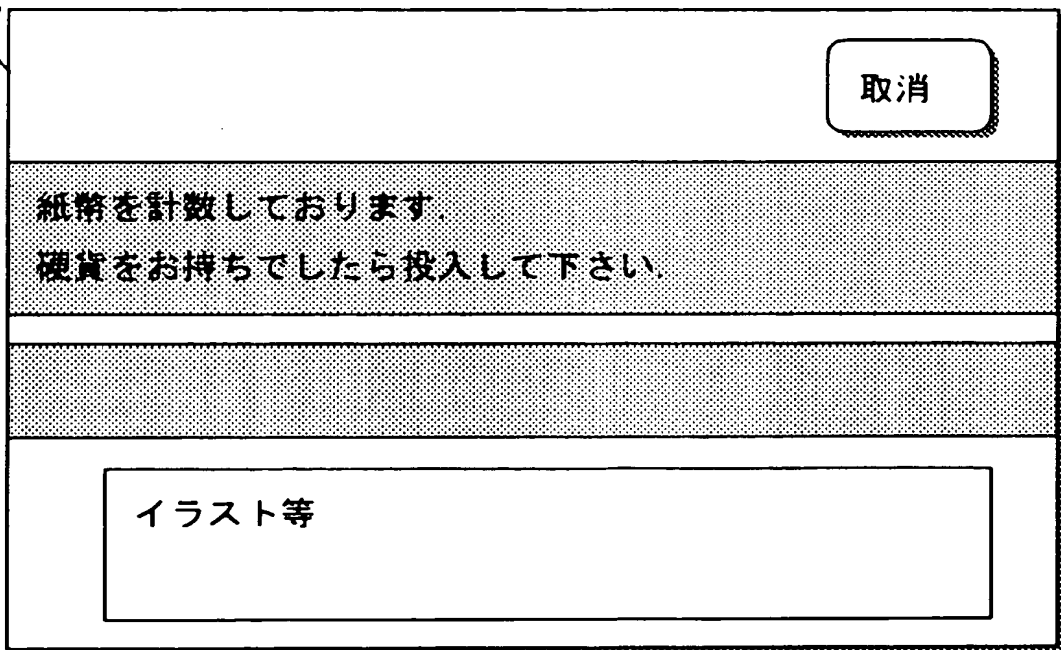
取消

紙幣と硬貨を入れてください。
紙幣は xxx 枚まで硬貨は xxx 枚まで挿入できます。

イラスト等



②紙幣挿入画面



取消

紙幣を計数しております。
硬貨をお持ちでしたら投入して下さい。

イラスト等

【図 15】

③硬貨挿入画面

取消

硬貨を計数しております。
紙幣をお持ちでしたら挿入して下さい。

イラスト等



④紙幣/硬貨挿入画面

取消

紙幣と硬貨を計数しております。
しばらくお待ち下さい。

イラスト等



⑤追加挿入画面

取消

金額をご確認のうえ、紙幣と硬貨をお入れ下さい。
よろしければ確認ボタンを押して下さい。

合計入金金額 xxxxxx 円

イラスト等

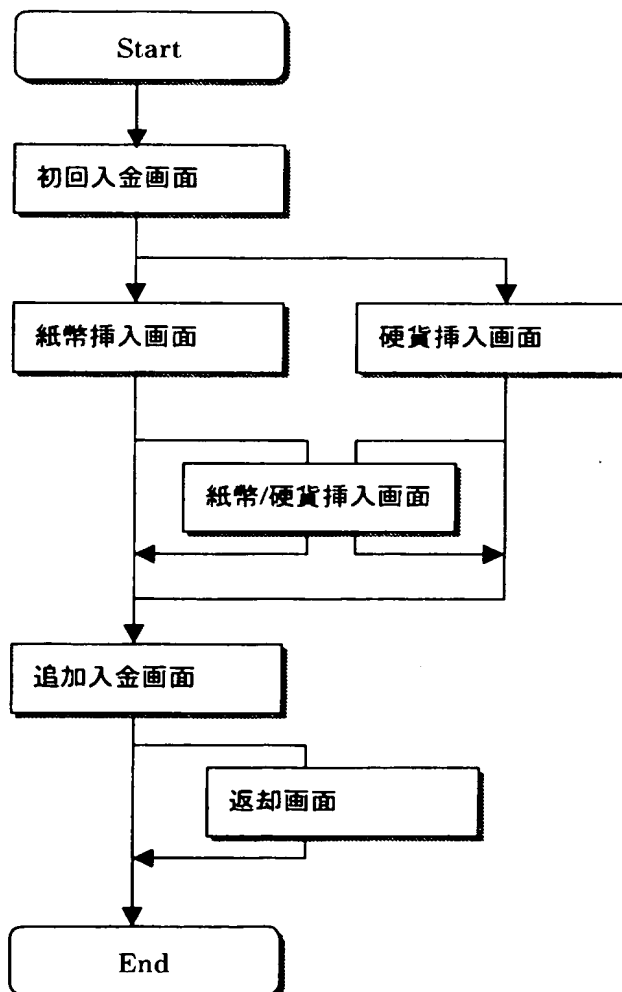
確認

【図 16】

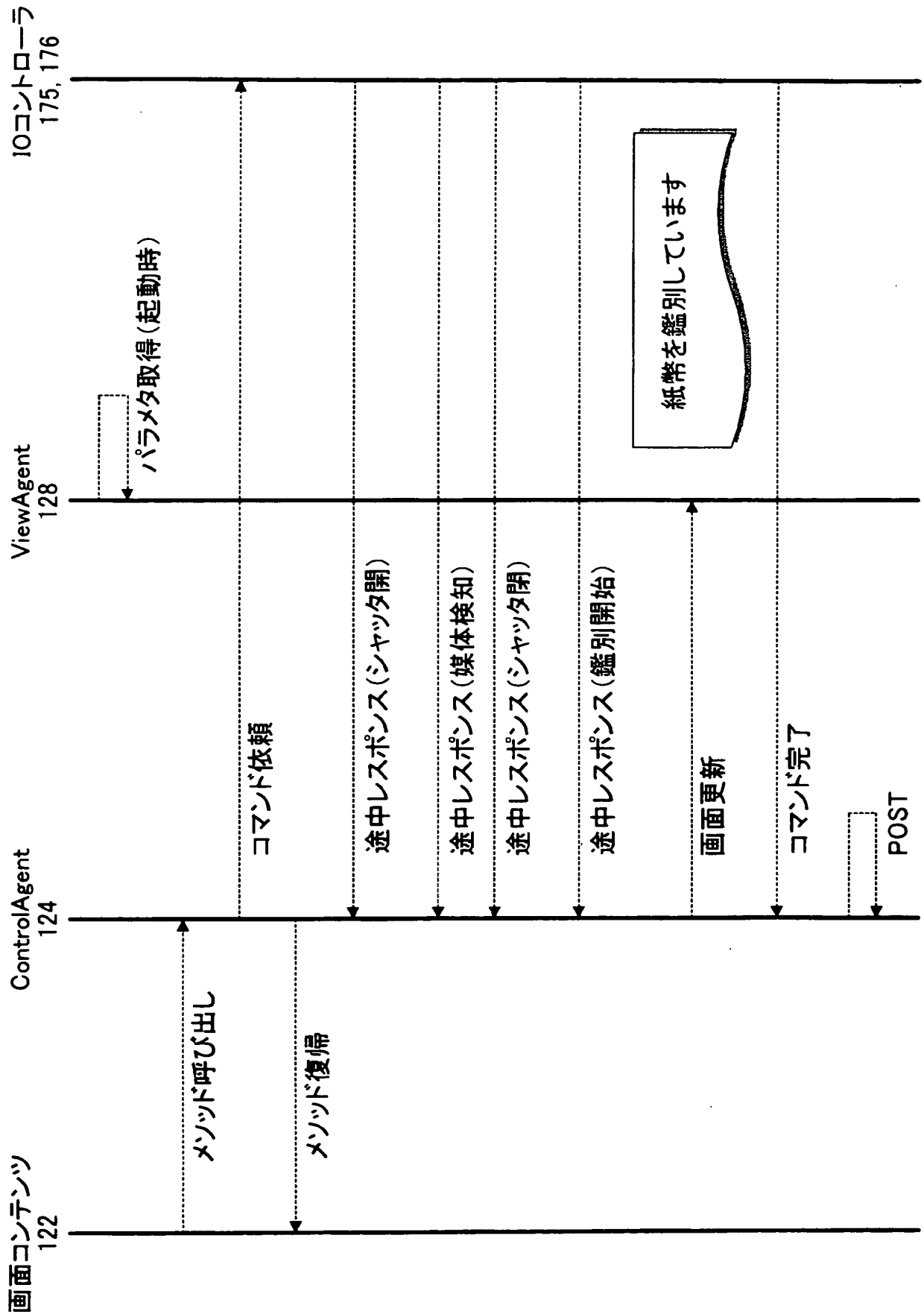
⑥鑑別不良返却画面

取消	
現金をお返しします 現金抜き取り後、シャッターが閉じます	
合計入金金額	XXXXXX 円
イラスト等	

【図 17】



【図 18】



【図 19】

```

<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html; charset=shift_jis">
<TITLE>WebATM Sample</TITLE>
</HEAD>
<BODY background="http://webatm:8080/webatm/xxx.bmp">

// 画面内で呼び出す Agent を指定する.
<APPLET CODE="U_agtSync_insertCash.insertCash" CODEBASE = "http://webatm:8080/webatm"
      WIDTH=0 HEIGHT=0 NAME="U_agtSync_insertCash" MAYSCRIPT>
</applet>

// POSTAgent が JavaScript の関数を呼び出せるように MAYSCRIPT 定義を行う.
<APPLET CODE="U_agtPost" CODEBASE = "http://webatm:8080/webatm"
      WIDTH=0 HEIGHT=0 NAME="U_agtPost" MAYSCRIPT>
</applet>

// ViewAgent を指定する.
<APPLET CODE="U_agtView" CODEBASE = "http://webatm:8080/webatm"
      WIDTH=100 HEIGHT=100 ALIGN=LEFT HSAPCE=10 VSPACE=10 NAME="U_agtView" >
  <PARAM NAME=fontKind  VALUE="Serif">
  <PARAM NAME=fontPoint VALUE="14">
  <PARAM NAME=fontStyle  VALUE="Bold">
  <PARAM NAME=fontColor  VALUE="Red">
  <PARAM NAME=fontColor  VALUE="Red">
  <PARAM NAME=background VALUE="Black">
  <PARAM NAME=startX     VALUE="10">
  <PARAM NAME=startY     VALUE="10">
  <PARAM NAME=endX       VALUE="100">
  <PARAM NAME=endY       VALUE="80">
  <PARAM NAME=text       VALUE="紙幣を鑑別しています。">
</applet>
} ViewAgent のパラメタ定義

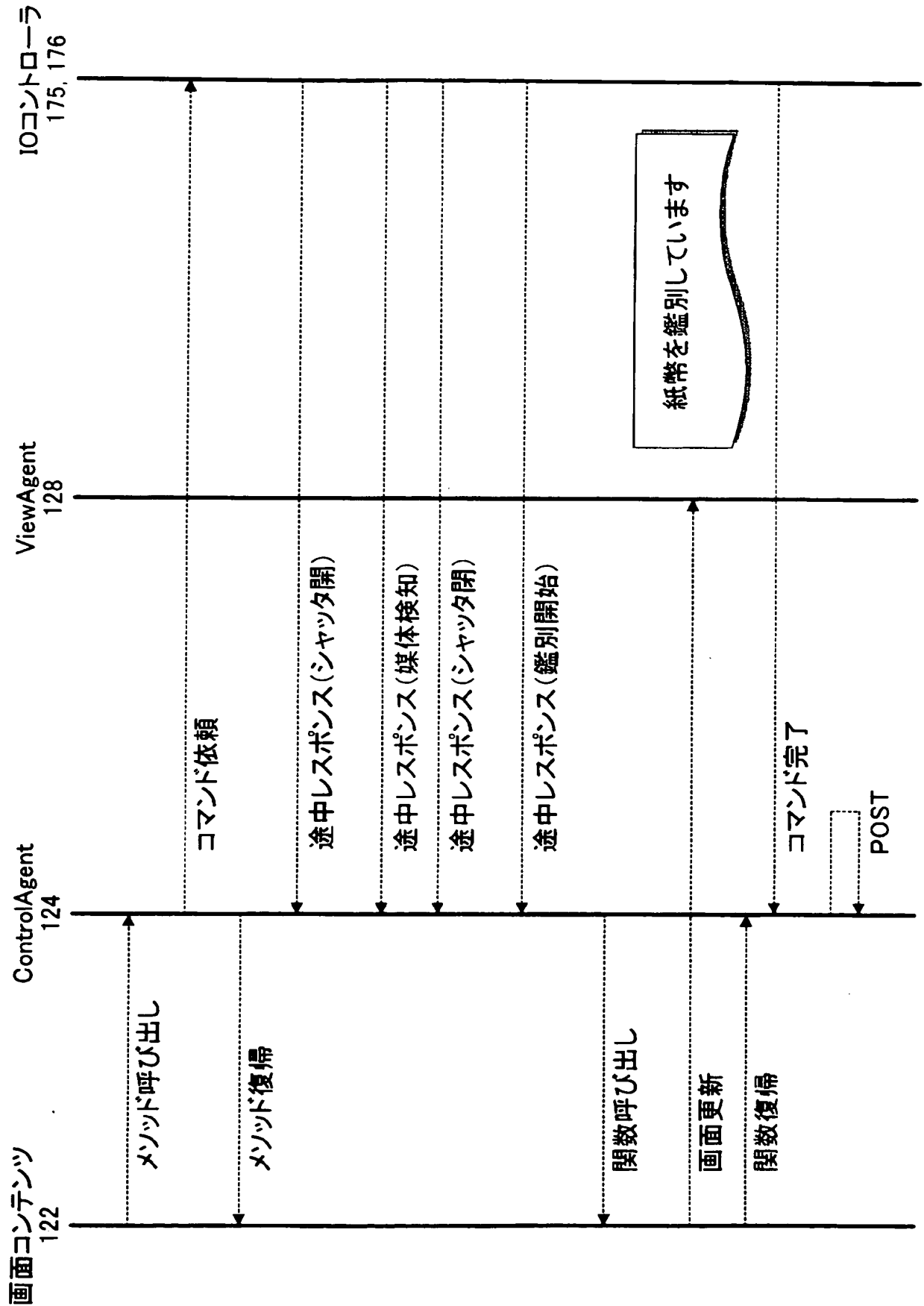
<SCRIPT language="javascript">
<!--
  // POSTAgent から呼び出される関数を定義する。(必須)
  function postFunc(postData){
    document.sys_form.ioResponse.value = postData;
    window.document.sys_form.submit();
  }

// -->
</SCRIPT>
<FORM name="sys_form" method="post" action="http://webatm:8080/webatm/main.jsp">
  ~省略~
  <font size=10>紙幣/硬貨を挿入して下さい。</font>
  <BR>
  <HR>
</FORM>
<SCRIPT language="javascript">
<!--
  // ControlAgent の呼び出し
  ret = document.U_agtSync_insertCash.insertCash(insertCashMode, inputParam, postMode, maxTimer);
  if (ret < 0) {
    // 異常通知処理
    ~省略~
  }

// -->
</SCRIPT>
</BODY>
</HTML>

```

【図 20】



【図 2 1】

```

<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html; charset=shift_jis">
<TITLE>WebATM Sample</TITLE>
</HEAD>
<BODY background="http://webatm:8080/webatm/xxx.bmp">

// 画面内で呼び出す Agent を指定する.
<applet code="U_agtSync_insertCash.insertCash" codebase = "http://webatm:8080/webatm"
width=0 height=0 name="U_agtSync_insertCash" MAYSCRIPT>
</applet>

// POSTAgent が JavaScript の関数を呼び出せるように MAYSCRIPT 定義を行う.
<applet code="U_agtPost" codebase = "http://webatm:8080/webatm"
width=0 height=0 name="U_agtPost" MAYSCRIPT>
</applet>

// ViewAgent を指定する.
<applet code="U_agtView" codebase = "http://webatm:8080/webatm"
width=100 height=100 align=LEFT hspace=10 vspace=10 name="U_agtView" >
</applet>

<SCRIPT language="javascript">
<!--
    // POSTAgent から呼び出される関数を定義する. (必須)
    function postFunc(postData){
        document.sys_form.ioResponse.value = postData;
        window.document.sys_form.submit();
    }

    // ControlAgent から呼び出される関数を定義する. (必須)
    function insertCashFunc(eventKind, billVal, coinVal){
        switch(eventKind) {
            case 1: // 鑑別開始 (紙幣)
                // ViewAgent の呼び出し
                ret = document.U_agtView.setFont(fontKind, fontPoint, fontStyle, fontColor, background);
                ret = document.U_agtView.showText(startX, startY, endX, endY, text);
                break;
            case 2: // 鑑別開始 (硬貨)
                ~省略~
            case n: // 鑑別終了 (紙幣/硬貨)
                ~省略~
        }
    }
// -->
</SCRIPT>
<FORM name="sys_form" method="post" action="http://webatm:8080/webatm/main.jsp">
    ~省略~
    <font size=10>紙幣/硬貨を挿入して下さい。</font>
    <BR>
    <HR>
</FORM>
<SCRIPT language="javascript">
<!--
    // ControlAgent の呼び出し
    ret = document.U_agtSync_insertCash.insertCash(insertCashMode, inputParam, postMode, maxTimer);
    if (ret < 0) {
        // 異常通知処理
        ~省略~
    }
// -->
</SCRIPT>
</BODY>
</HTML>

```

【図 2 2】

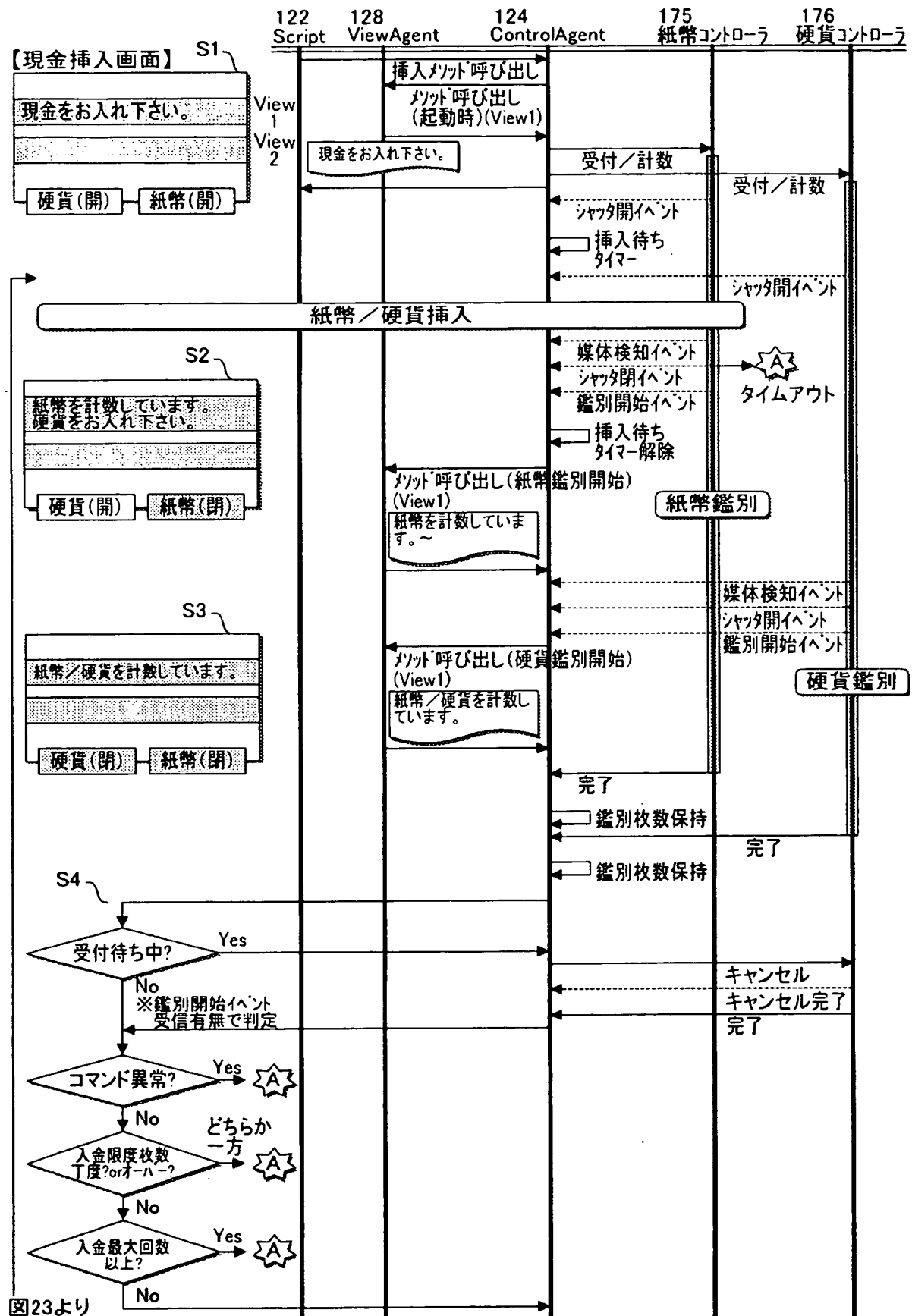
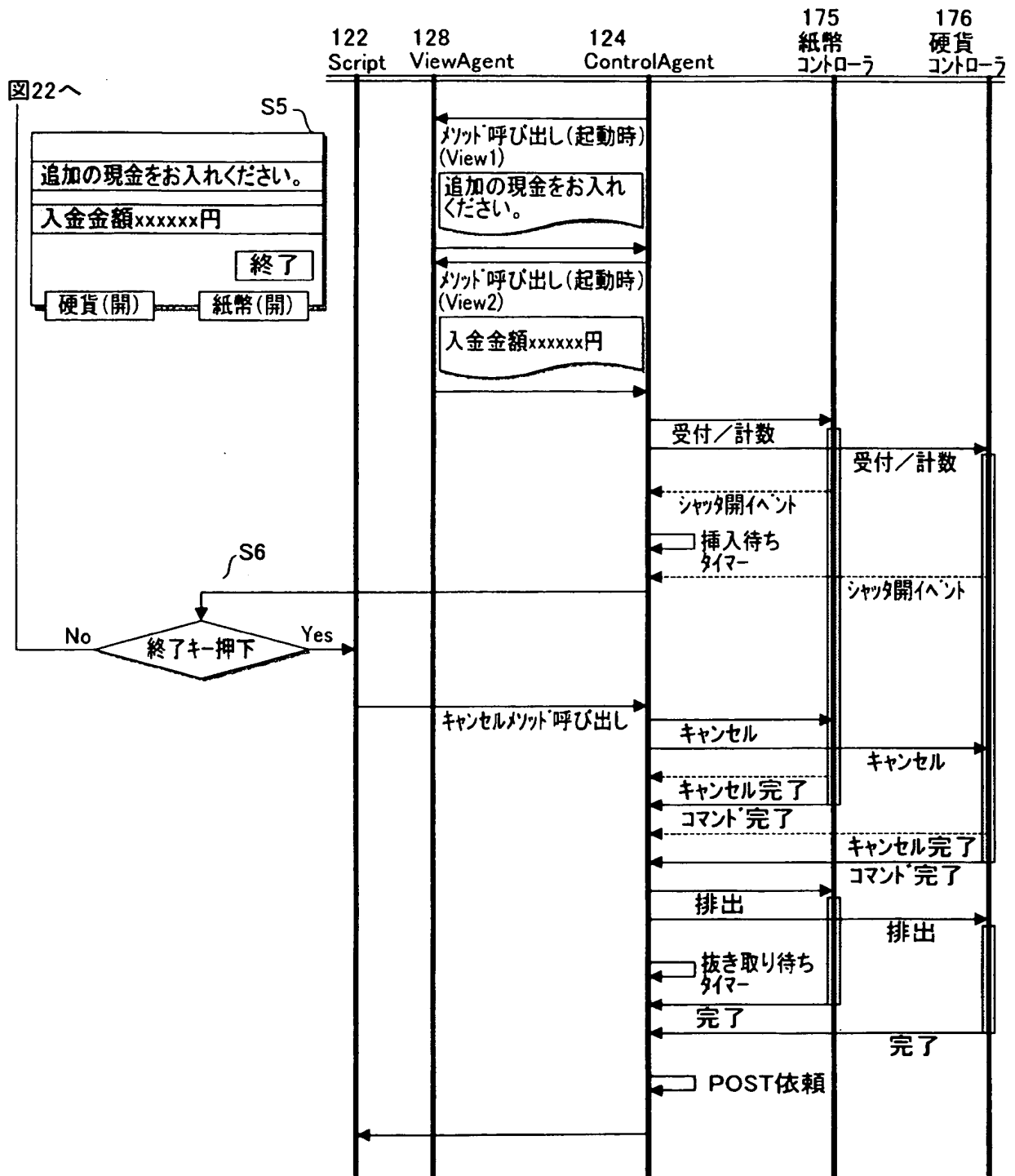
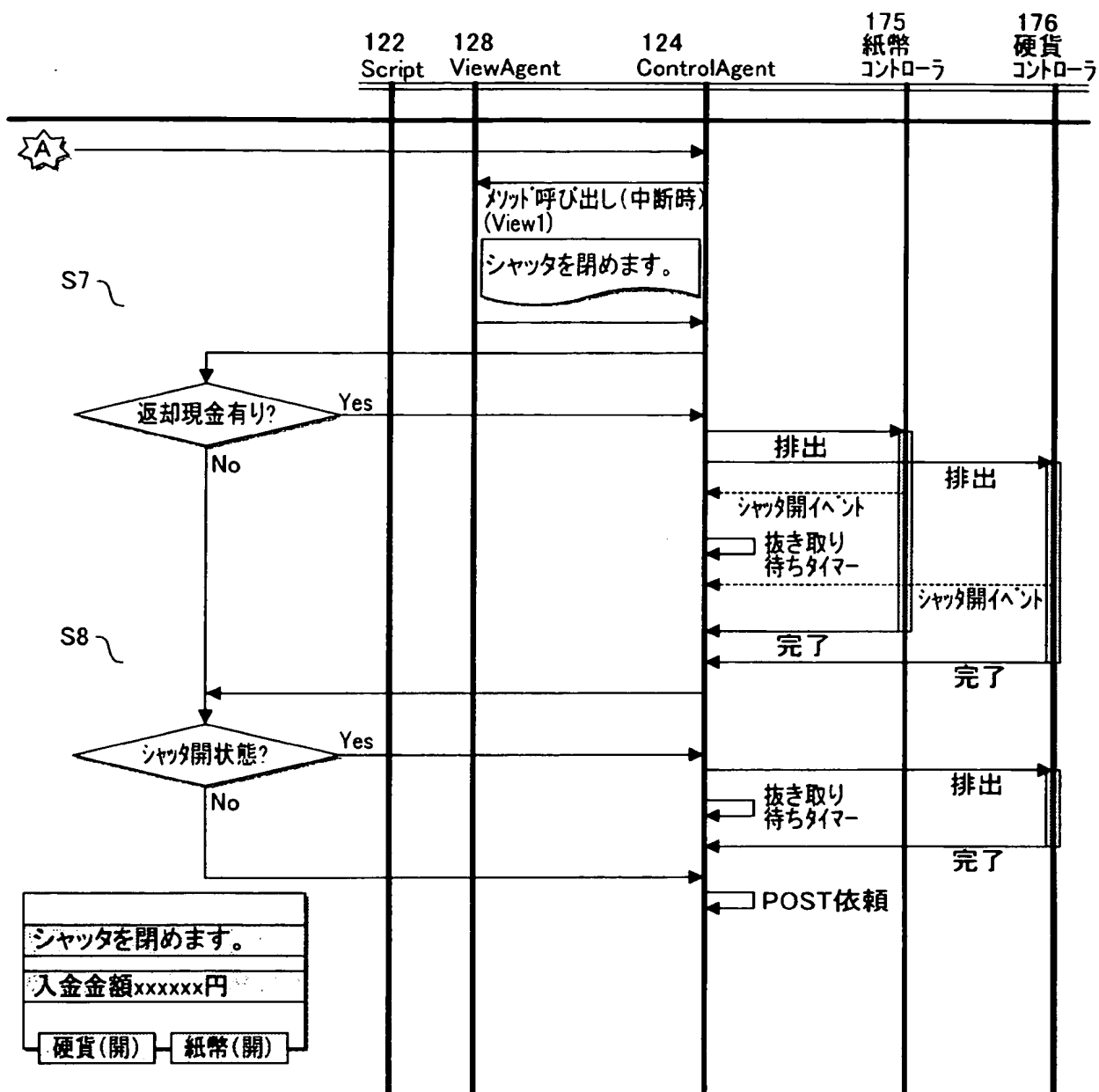


図23より

【図 23】



【図 24】



【図 25】

Agent	method	対象IOコントローラ								
		なし	紙幣	硬貨	通帳	カード	レシート	ジャーナル	取引	テンキー
同期制御	初期化		●	●	●	●	●	●	●	●
	メカリセット		●	●	●	●	●	●		
	紙幣／硬貨挿入		●	●						
	媒体同時排出		●	●						
	印字／繰出／MSライイト／排出準備		●	●	●	●	●	●		
	入金返却		●	●	●	●	●			
	収納		●	●						
	強制排出／取込み		●	●	●	●	●			
	ユニット情報取得／取引状態設定／二面表示制御		●	●	●	●	●	●	●	●
	入出金準備		●	●						
POST	強制補充		●	●						
	ジャムリセット		●	●						
	カード／通帳挿入				●	●				
	POST処理	●								
テキスト表示	POSTデータ保持	●								
	フォント設定	●								
	テキスト表示	●								
	テキスト消去	●								

【図 26】

```

<HTML>
<HEAD>
<TITLE>WebATM Sample</TITLE>
</HEAD>

<BODY>

// 画面内で呼び出す Agent を指定する.
<APPLET CODE="U_agtSync.class" codebase = "http://webatm:8080/webatm"
        WIDTH="0" HEIGHT="0" NAME="U_agtSync" >
</APPLET>

// POSTAgent が JavaScript の関数を呼び出せるように MAYSCRIPT 定義を行う.
<APPLET CODE="U_agtPost.class" codebase = "http://webatm:8080/webatm"
        WIDTH="0" HEIGHT="0" NAME="U_agtPost" MAYSCRIPT>

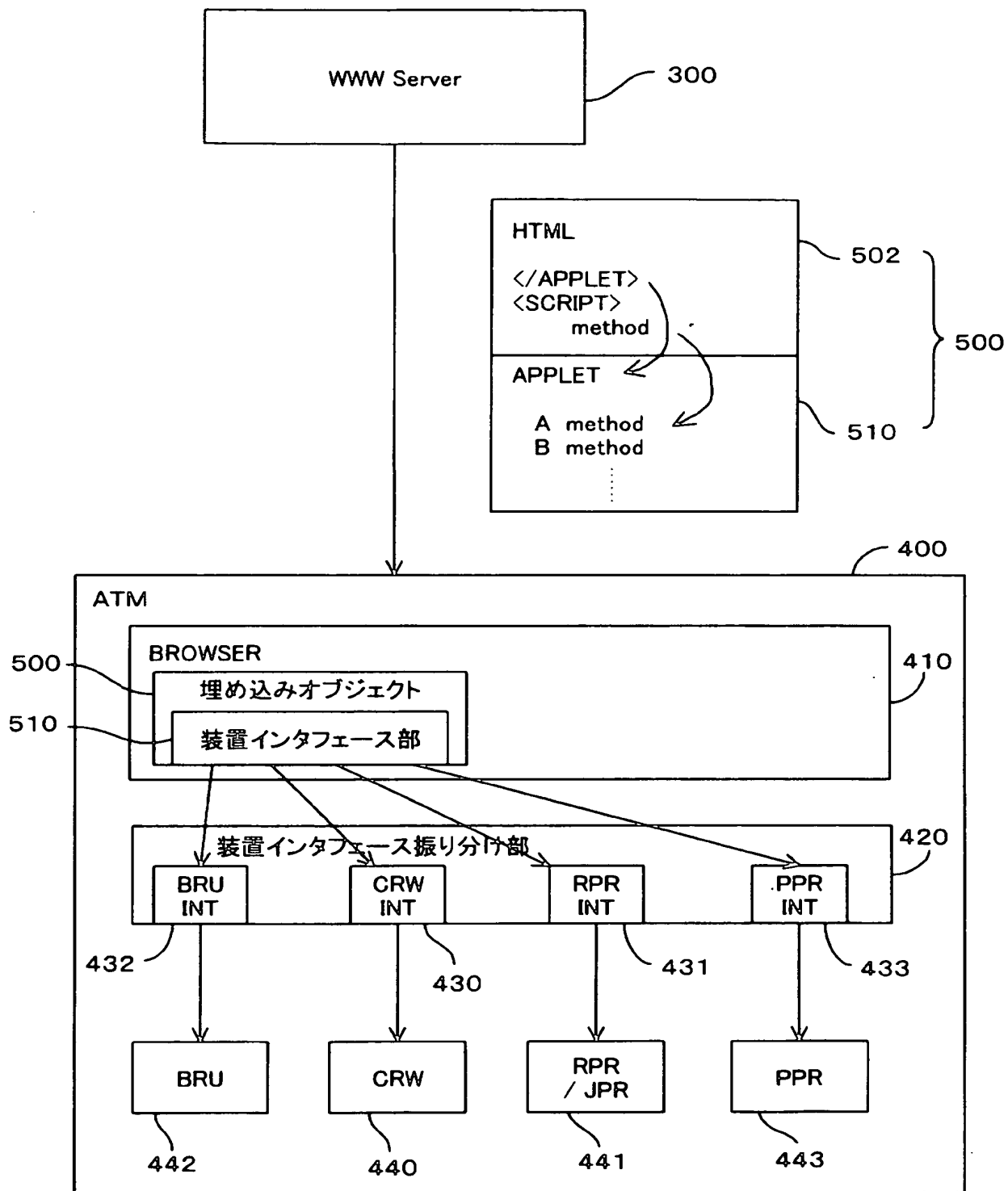
<SCRIPT language="javascript">
<!--
// POSTAgent から呼び出される関数を定義する. (必須)
function postFunc(postData){ document.sys_form.ioResponse.value = postData;
        window.document.sys_form.submit();
    }

// 同期 Agent の初期化メソッドの呼び出し
ret = document.U_agtSync_initial (inputParam, parseInt(postMode), parseInt(maxTimer));
if (ret < 0) {
    // 異常通知処理
    ~省略~
}

// -->
</SCRIPT>
</BODY>
</HTML>

```

【図 27】





【図 28】

<HTML>

(表示画像のプログラム記述)省略

<applet code=MachineIF.class codebase=MachineIF id=IDMachinesIF>

</applet> (D1)

<script>

// 各装置を初期化する。

FunctionInitialize{

IDMachineIF. Initialize("Cash");

IDMachineIF. Initialize("Card");

IDMachineIF. Initialize("Receipt");

IDMachineIF. Initialize("Passbook");

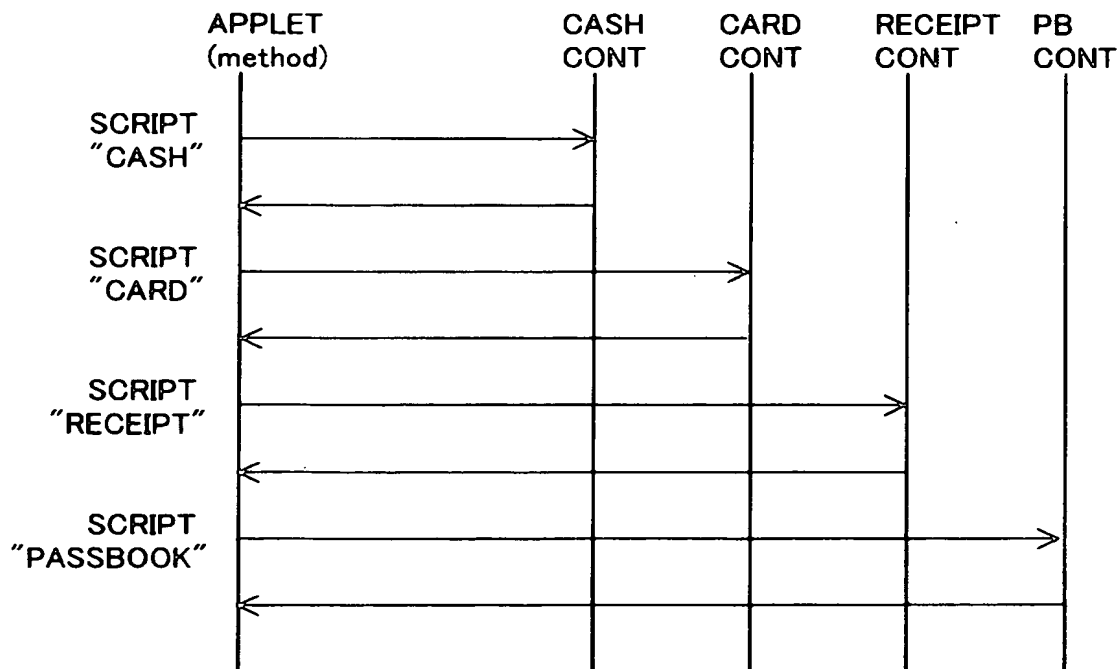
}

</script>

(表示画像のプログラム記述)省略

</HTML>

【図 29】



【書類名】 要約書**【要約】**

【課題】 利用者の操作に応じて、W e b サーバと交信し、自動取引する自動取引装置に関し、W e b サーバへのリクエスト回数を削減して、自動取引処理する。

【解決手段】 W e b サーバ（1 0 0）の画面コンテンツ（1 2 2）に取引処理の各処理単位のオブジェクト（1 2 4）を埋め込み、画面の更新を全てW e b サーバからダウンロードするのではなく、画面コンテンツのアプレットとスクリプトで、画面の一部を更新する。W e b サーバとの通信を行わずに、画面の更新が可能となる。このため、W e b サーバ 1 0 0 へのリクエスト回数が減少し、通信回数及び通信時間を減少でき、自動取引装置の取引性能（速度）を向上でき、且つW e b サーバの負荷を軽減できる。

【選択図】 図 1 8

特願 2 0 0 3 - 3 9 0 4 7 7

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 5 2 2 3]

1. 変更年月日 1 9 9 6 年 3 月 2 6 日

[変更理由] 住所変更

住 所 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号

氏 名 富士通株式会社

特願 2 0 0 3 - 3 9 0 4 7 7

出 願 人 履 歴 情 報

識別番号

[0 0 0 2 3 7 6 3 9]

1. 変更年月日

2 0 0 2 年 7 月 9 日

[変更理由]

名称変更

住 所

東京都稲城市矢野口 1 7 7 6 番地

氏 名

富士通フロンテック株式会社